# Open Source Used In SamKnows SDK Android 2.3.15

**Cisco Systems, Inc.**
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Text Part Number: 78EE117C99-2098071122

**This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please submit this form.**

**In your requests please include the following reference number 78EE117C99-2098071122**

# Contents


**1.1 trove4j 1.0.20181211**
1.1.1 Available under license
**1.2 automation 1.11-8**
**1.3 kxml 2.3.0**
1.3.1 Available under license
**1.4 juniversalchardet 1.0.3**
1.4.1 Available under license
**1.5 txw2 2.3.2**
1.5.1 Available under license
**1.6 jakarta xml bind api 2.3.2**
1.6.1 Available under license
**1.7 jakarta activation api 1.2.1**
1.7.1 Available under license
**1.8 generex 1.0.2**
1.8.1 Available under license
**1.9 commons-logging 1.2**
1.9.1 Available under license
**1.10 commons-lang3 3.7**
1.10.1 Available under license
**1.11 gson 2.8.0**
1.11.1 Available under license
**1.12 json-simple 1.1**
1.12.1 Available under license
**1.13 antlr-runtime 4.5.3**
1.13.1 Available under license
**1.14 j2objc-annotations 1.3**

1.14.1 Available under license
**1.15 javax-activation 1.2.0**
1.15.1 Available under license
**1.16 jaxb-runtime 2.3.2**
1.16.1 Available under license
**1.17 stax-ex 1.8.1**
1.17.1 Available under license
**1.18 antlr-runtime 4.7.1**
**1.19 auto-value-annotations 1.6.2**
**1.20 rxjava 2.2.6**
**1.21 joda-time 2.10.3**
1.21.1 Available under license
**1.22 annotations 4.1.1.4**
**1.23 commonmark 0.11.0**
**1.24 jmustache 1.14**
**1.25 rhino 1.7.7.2**
1.25.1 Available under license
**1.26 converter-scalars 2.5.0**
**1.27 converter-gson 2.5.0**
**1.28 handlebars-jackson2 4.1.2**
**1.29 threetenbp 1.4.0**
1.29.1 Available under license
**1.30 listenablefuture 1.0**
1.30.1 Available under license
**1.31 jakarta-validation-api 2.0.2**
1.31.1 Available under license
**1.32 reactive-streams 1.0.3**
1.32.1 Available under license
**1.33 javac-shaded 9-dev-r4023-3**
1.33.1 Available under license
**1.34 google-java-format 1.5**
1.34.1 Available under license
**1.35 slf4j-simple 1.7.29**
1.35.1 Available under license
**1.36 slf4j-ext 1.7.29**
1.36.1 Available under license
**1.37 slf4j 1.7.30**
1.37.1 Available under license
**1.38 commons-compress 1.20**

1.38.1 Available under license
**1.39 fast-infoset 1.2.16**
1.39.1 Available under license
**1.40 checker-qual 2.11.1**
1.40.1 Available under license
**1.41 caffeine 2.8.1**
1.41.1 Available under license
**1.42 httpcore 4.4.13**
1.42.1 Available under license
**1.43 javassist 3.26.0-GA**
1.43.1 Available under license
**1.44 jdom 2.0.6**
1.44.1 Available under license
**1.45 retrofit 2.9.0**
1.45.1 Available under license
**1.46 swagger-core 1.6.2**
1.46.1 Available under license
**1.47 swagger-annotations 1.6.2**
1.47.1 Available under license
**1.48 netty-resolver 4.1.52.Final**
1.48.1 Available under license
**1.49 netty-codec 4.1.52.Final**
1.49.1 Available under license
**1.50 netty-handler 4.1.52.Final**
1.50.1 Available under license
**1.51 netty-transport 4.1.52.Final**
1.51.1 Available under license
**1.52 netty-buffer 4.1.52.Final**
1.52.1 Available under license
**1.53 checker-qual 3.5.0**
1.53.1 Available under license
**1.54 netty-handler-proxy 4.1.52.Final**
1.54.1 Available under license
**1.55 netty-codec-socks 4.1.52.Final**
1.55.1 Available under license
**1.56 jna-platform 5.6.0**
1.56.1 Available under license
**1.57 jna 5.6.0**
1.57.1 Available under license

**1.58 annotations 20.1.0**
1.58.1 Available under license
**1.59 httpclient 4.5.13**
1.59.1 Available under license
**1.60 joda-time 2.10.8**
1.60.1 Available under license
**1.61 xerces-impl 2.12.0**
1.61.1 Available under license
**1.62 guava 29.0-jre**
1.62.1 Available under license
**1.63 jsoup 1.13.1**
1.63.1 Available under license
**1.64 json 20180813**
1.64.1 Available under license
**1.65 bean-validation-api 1.1.0.Final**
1.65.1 Available under license
**1.66 msg-simple 1.2**
1.66.1 Available under license
**1.67 btf 1.3**
1.67.1 Available under license
**1.68 joda-time 2.10.9**
1.68.1 Available under license
**1.69 guava 30.1-jre**
1.69.1 Available under license
**1.70 jackson-datatype-threetenbp 2.10.0**
1.70.1 Available under license
**1.71 proto-google-common-protos 2.0.1**
1.71.1 Available under license
**1.72 commons-io 2.6**
1.72.1 Available under license
**1.73 checker-compat-qual 2.5.5**
1.73.1 Available under license
**1.74 annotations 13.0**
1.74.1 Available under license
**1.75 rgxgen 1.3**
1.75.1 Available under license
**1.76 mailapi 1.6.2**
1.76.1 Available under license
**1.77 libphonenumber 8.11.1**

1.77.1 Available under license
**1.78 jctools-core 3.1.0**
1.78.1 Available under license
**1.79 commons-lang3 3.8.1**
1.79.1 Available under license
**1.80 commons-codec 1.11**
1.80.1 Available under license
**1.81 commons-cli 1.2**
1.81.1 Available under license
**1.82 swagger-annotations 2.1.7**
1.82.1 Available under license
**1.83 swagger-models 2.1.7**
1.83.1 Available under license
**1.84 jackson-coreutils-equivalence 1.0**
1.84.1 Available under license
**1.85 json-schema-core 1.2.14**
1.85.1 Available under license
**1.86 uritemplate 0.10**
1.86.1 Available under license
**1.87 json-schema-validator 2.2.14**
1.87.1 Available under license
**1.88 asm 9.1**
1.88.1 Available under license
**1.89 swagger-core 2.1.7**
1.89.1 Available under license
**1.90 gson 2.8.6**
1.90.1 Available under license
**1.91 xsdlib 2013.6.1**
1.91.1 Available under license
**1.92 jopt-simple 5.0.4**
1.92.1 Available under license
**1.93 jackson-annotations 2.12.4**
1.93.1 Available under license
**1.94 jackson-datatype-jsr310 2.12.4**
1.94.1 Available under license
**1.95 swagger-parser-core 2.0.26**
1.95.1 Available under license
**1.96 swagger-parser-v3 2.0.26**
1.96.1 Available under license

**1.97 jackson-dataformat-yaml 2.12.4**
1.97.1 Available under license
**1.98 jackson-xc 2.12.4**
1.98.1 Available under license
**1.99 istack-commons-runtime 3.0.8**
1.99.1 Available under license
**1.100 asm-util 9.1**
1.100.1 Available under license
**1.101 asm-tree 9.1**
1.101.1 Available under license
**1.102 asm-commons 9.1**
1.102.1 Available under license
**1.103 asm-analysis 9.1**
1.103.1 Available under license
**1.104 protobuf-java-util 3.17.2**
1.104.1 Available under license
**1.105 protobuf-java 3.17.2**
1.105.1 Available under license
**1.106 error_prone_annotations 2.4.0**
1.106.1 Available under license
**1.107 animal-sniffer-annotation 1.19**
1.107.1 Available under license
**1.108 error_prone_annotations 2.7.1**
1.108.1 Available under license
**1.109 jackson-datatype-guava 2.12.4**
1.109.1 Available under license
**1.110 jackson-datatype-joda 2.12.4**
1.110.1 Available under license
**1.111 bcprov-jdk15on 1.67**
1.111.1 Available under license
**1.112 bcpkix-jdk15on 1.67**
1.112.1 Available under license
**1.113 guava 31.0.1-jre**
1.113.1 Available under license
**1.114 grpc-netty 1.39.0**
1.114.1 Available under license
**1.115 libcxx 9.0.8svn**
1.115.1 Available under license
**1.116 libcxxabi 9.0.8svn**

1.116.1 Available under license
**1.117 swagger-compat-spec-parser 1.0.55**
1.117.1 Available under license
**1.118 swagger-parser 1.0.55**
1.118.1 Available under license
**1.119 swagger-parser-v2-converter 2.0.26**
1.119.1 Available under license
**1.120 commons-compress 1.21**
1.120.1 Available under license
**1.121 checker-qual 3.12.0**
1.121.1 Available under license
**1.122 stax-api 4.2.1**
1.122.1 Available under license
**1.123 hawtjni-runtime 1.15**
1.123.1 Available under license
**1.124 jansi-native 1.7**
1.124.1 Available under license
**1.125 jansi-osx 1.7**
1.125.1 Available under license
**1.126 jansi 1.16**
1.126.1 Available under license
**1.127 jansi 1.7**
1.127.1 Available under license
**1.128 gson 2.9.0**
1.128.1 Available under license
**1.129 trove4j 1.0.20200330**
1.129.1 Available under license
**1.130 jackson-core 2.12.4**
1.130.1 Available under license
**1.131 perfmark-api 0.23.0**
1.131.1 Available under license
**1.132 grpc-api 1.39.0**
1.132.1 Available under license
**1.133 grpc-core 1.39.0**
1.133.1 Available under license
**1.134 grpc-stub 1.39.0**
1.134.1 Available under license
**1.135 grpc-context 1.39.0**
1.135.1 Available under license

**1.136 grpc-protobuf-lite 1.39.0**
1.136.1 Available under license
**1.137 grpc-protobuf 1.39.0**
1.137.1 Available under license
**1.138 kotlinx-coroutines-core-jvm 1.5.0**
1.138.1 Available under license
**1.139 failureaccess 1.0.1**
1.139.1 Available under license
**1.140 error_prone_annotations 2.3.4**
1.140.1 Available under license
**1.141 listenablefuture 9999.0-empty-to-avoid-conflict-with-guava**
1.141.1 Available under license
**1.142 adapter-rxjava3 2.9.0**
1.142.1 Available under license
**1.143 logging-interceptor 4.10.0**
1.143.1 Available under license
**1.144 flatbuffers-java 1.12.0**
1.144.1 Available under license
**1.145 gson 2.8.9**
1.145.1 Available under license
**1.146 jimfs 1.1**
1.146.1 Available under license
**1.147 okhttp 4.10.0**
1.147.1 Available under license
**1.148 jose4j 0.7.0**
1.148.1 Available under license
**1.149 okio-jvm 3.0.0**
1.149.1 Available under license
**1.150 xz 1.9**
1.150.1 Available under license
**1.151 kotlin-stdlib-jdk8 1.5.31**
1.151.1 Available under license
**1.152 kotlin-stdlib-jdk7 1.5.31**
1.152.1 Available under license
**1.153 jsonpatch 1.13**
1.153.1 Available under license
**1.154 jackson-coreutils 2.0**
1.154.1 Available under license
**1.155 swagger-models 1.6.2**

1.155.1 Available under license
**1.156 kotlin-stdlib-common 1.5.31**
1.156.1 Available under license
**1.157 kotlinx-metadata-jvm 0.5.0**
1.157.1 Available under license
**1.158 woodstox-core 6.2.4**
1.158.1 Available under license
**1.159 javax-annotation-api 1.3.2**
1.159.1 Available under license
**1.160 isorelax 20090621**
1.160.1 Available under license
**1.161 jsr305 3.0.2**
1.161.1 Available under license
**1.162 xml-apis 1.4.01**
1.162.1 Available under license
**1.163 aapt2 7.3.0-8691043-osx**
1.163.1 Available under license
**1.164 layoutlib-api 30.3.0**
1.164.1 Available under license
**1.165 kotlin-util-io 1.7.10**
1.165.1 Available under license
**1.166 kotlin-build-common 1.7.10**
1.166.1 Available under license
**1.167 android-test-plugin-result-listener-gradle-proto 30.3.0**
1.167.1 Available under license
**1.168 lifecycle-common 2.0.0**
1.168.1 Available under license
**1.169 antlr-runtime 4.5.2-1**
1.169.1 Available under license
**1.170 lint-typedef-remover 30.3.0**
1.170.1 Available under license
**1.171 kotlin-klib-commonizer-api 1.7.10**
1.171.1 Available under license
**1.172 kotlin-scripting-compiler-impl-embeddable 1.5.31**
1.172.1 Available under license
**1.173 javawriter 2.5.0**
1.173.1 Available under license
**1.174 common 30.3.0**
1.174.1 Available under license

**1.175 dagger 2.44**
1.175.1 Available under license
**1.176 kotlinx-coroutines-android 1.5.0**
1.176.1 Available under license
**1.177 aaptcompiler 7.3.0**
1.177.1 Available under license
**1.178 openapi-generator-gradle-plugin 5.2.1**
1.178.1 Available under license
**1.179 semver4j 0.16.4**
1.179.1 Available under license
**1.180 kotlin-compiler-embeddable 1.7.10**
1.180.1 Available under license
**1.181 lifecycle-common 2.3.1**
1.181.1 Available under license
**1.182 kotlin-compiler-runner 1.7.10**
1.182.1 Available under license
**1.183 kotlin-annotation-processing-gradle 1.7.10**
1.183.1 Available under license
**1.184 kotlin-gradle-plugin 1.7.10-gradle70**
1.184.1 Available under license
**1.185 kotlin-native-utils 1.7.10**
1.185.1 Available under license
**1.186 lint-model 30.3.0**
1.186.1 Available under license
**1.187 kotlin-native-utils 1.5.31**
1.187.1 Available under license
**1.188 repository 30.3.0**
1.188.1 Available under license
**1.189 transform-api 2.0.0-deprecated-use-gradle-api**
1.189.1 Available under license
**1.190 kotlin-daemon-client 1.7.10**
1.190.1 Available under license
**1.191 gradle-api 7.3.0**
1.191.1 Available under license
**1.192 kotlin-android-extensions 1.5.31**
1.192.1 Available under license
**1.193 kotlin-annotation-processing-gradle 1.5.31**
1.193.1 Available under license
**1.194 builder-model 7.3.0**

1.194.1 Available under license

**1.195 dagger-compiler 2.44**

1.195.1 Available under license

**1.196 kotlin-stdlib-jdk7 1.5.0**

1.196.1 Available under license

**1.197 android-test-plugin-host-additional-test-output-proto 30.3.0**

1.197.1 Available under license

**1.198 apkzlib 7.3.0**

1.198.1 Available under license

**1.199 kotlin-tooling-metadata 1.5.31**

1.199.1 Available under license

**1.200 bundletool 1.9.0**

1.200.1 Available under license

**1.201 kotlin-stdlib 1.8.0**

1.201.1 Available under license

**1.202 kotlin-project-model 1.7.10**

1.202.1 Available under license

**1.203 kotlinx-coroutines-android 1.6.1**

1.203.1 Available under license

**1.204 jetifier-processor 1.0.0-beta10**

1.204.1 Available under license

**1.205 annotation 1.2.0**

1.205.1 Available under license

**1.206 kotlin-stdlib-jdk7 1.8.0**

1.206.1 Available under license

**1.207 kotlin-stdlib-jdk8 1.8.0**

1.207.1 Available under license

**1.208 databinding-compiler-common 7.3.0**

1.208.1 Available under license

**1.209 databinding-common 7.3.0**

1.209.1 Available under license

**1.210 kotlin-gradle-plugin-model 1.7.10**

1.210.1 Available under license

**1.211 symbol-processing-api 1.7.0-1.0.6**

1.211.1 Available under license

**1.212 tensorflow-lite-metadata 0.1.0-rc2**

1.212.1 Available under license

**1.213 kotlin-build-common 1.5.31**

1.213.1 Available under license

**1.214 kotlin-gradle-plugin-model 1.5.31**
1.214.1 Available under license
**1.215 dagger-producers 2.44**
1.215.1 Available under license
**1.216 aapt2-proto 7.3.0-8691043**
1.216.1 Available under license
**1.217 signflinger 7.3.0**
1.217.1 Available under license
**1.218 kotlin-klib-commonizer-api 1.5.31**
1.218.1 Available under license
**1.219 kotlin-stdlib-jdk8 1.7.0**
1.219.1 Available under license
**1.220 tink 1.3.0-rc2**
1.220.1 Available under license
**1.221 kotlin-sam-with-receiver 1.5.31**
1.221.1 Available under license
**1.222 rxjava 3.0.0**
1.222.1 Available under license
**1.223 hilt-android-gradle-plugin 2.44**
1.223.1 Available under license
**1.224 kotlin-scripting-compiler-impl-embeddable 1.7.10**
1.224.1 Available under license
**1.225 kotlin-script-runtime 1.5.31**
1.225.1 Available under license
**1.226 kotlin-scripting-jvm 1.5.31**
1.226.1 Available under license
**1.227 gradle-kotlin-dsl-plugins 2.1.7**
1.227.1 Available under license
**1.228 kotlin-android-extensions 1.7.10**
1.228.1 Available under license
**1.229 kotlin-reflect 1.8.0**
1.229.1 Available under license
**1.230 jackson-module-kotlin 2.12.4**
1.230.1 Available under license
**1.231 kotlin-stdlib 1.7.0**
1.231.1 Available under license
**1.232 protos 30.3.0**
1.232.1 Available under license
**1.233 tracker 30.3.0**

1.233.1 Available under license

**1.234 annotations 30.3.0**

1.234.1 Available under license

**1.235 sdklib 30.3.0**

1.235.1 Available under license

**1.236 collection 1.0.0**

1.236.1 Available under license

**1.237 kotlin-script-runtime 1.7.10**

1.237.1 Available under license

**1.238 kotlin-util-io 1.5.31**

1.238.1 Available under license

**1.239 kotlinx-coroutines-core-jvm 1.6.1**

1.239.1 Available under license

**1.240 kotlin-gradle-plugin-idea 1.7.10**

1.240.1 Available under license

**1.241 zipflinger 7.3.0**

1.241.1 Available under license

**1.242 kotlin-gradle-plugin-api 1.5.31**

1.242.1 Available under license

**1.243 gradle-test-logger-plugin 1.6.0**

1.243.1 Available under license

**1.244 kotlin-scripting-compiler-embeddable 1.7.10**

1.244.1 Available under license

**1.245 hilt-core 2.44**

1.245.1 Available under license

**1.246 fastutil 8.4.0**

1.246.1 Available under license

**1.247 concurrent-futures 1.0.0**

1.247.1 Available under license

**1.248 kotlin-tooling-metadata 1.7.10**

1.248.1 Available under license

**1.249 kotlin-tooling-core 1.7.10**

1.249.1 Available under license

**1.250 retrofit2-kotlin-coroutines-adapter 0.9.2**

1.250.1 Available under license

**1.251 openapi-generator 5.2.1**

1.251.1 Available under license

**1.252 dokka-core 1.5.30**

1.252.1 Available under license

**1.253 apksig 7.3.0**
1.253.1 Available under license
**1.254 core-common 2.0.0**
1.254.1 Available under license
**1.255 incap 0.2**
1.255.1 Available under license
**1.256 crash 30.3.0**
1.256.1 Available under license
**1.257 kotlin-stdlib-jdk7 1.7.10**
1.257.1 Available under license
**1.258 builder-test-api 7.3.0**
1.258.1 Available under license
**1.259 httpmime 4.5.6**
1.259.1 Available under license
**1.260 kotlin-stdlib-jdk8 1.7.10**
1.260.1 Available under license
**1.261 kotlin-stdlib-common 1.7.0**
1.261.1 Available under license
**1.262 kotlin-stdlib-common 1.8.0**
1.262.1 Available under license
**1.263 kotlin-daemon-embeddable 1.5.31**
1.263.1 Available under license
**1.264 kotlin-stdlib-jdk7 1.7.0**
1.264.1 Available under license
**1.265 lifecycle-common 2.5.1**
1.265.1 Available under license
**1.266 kotlin-stdlib-jdk8 1.5.0**
1.266.1 Available under license
**1.267 kotlin-gradle-plugin-api 1.7.10-gradle70**
1.267.1 Available under license
**1.268 kotlin-compiler-embeddable 1.5.31**
1.268.1 Available under license
**1.269 kotlin-stdlib-jdk7 1.6.10**
1.269.1 Available under license
**1.270 gradle-download-task 4.1.1**
1.270.1 Available under license
**1.271 constraintlayout-solver 1.1.3**
1.271.1 Available under license
**1.272 kotlin-stdlib 1.7.10**

1.272.1 Available under license

**1.273 kotlin-scripting-compiler-embeddable 1.5.31**

1.273.1 Available under license

**1.274 builder 7.3.0**

1.274.1 Available under license

**1.275 manifest-merger 30.3.0**

1.275.1 Available under license

**1.276 kotlin-reflect 1.5.31**

1.276.1 Available under license

**1.277 kotlinx-coroutines-core-jvm 1.5.1**

1.277.1 Available under license

**1.278 sdk-common 30.3.0**

1.278.1 Available under license

**1.279 jackson-dataformat-xml 2.12.4**

1.279.1 Available under license

**1.280 android-test-plugin-host-coverage-proto 30.3.0**

1.280.1 Available under license

**1.281 kotlinx-coroutines-jdk 1.5.0**

1.281.1 Available under license

**1.282 android-test-plugin-host-retention-proto 30.3.0**

1.282.1 Available under license

**1.283 swagger-parser 2.0.26**

1.283.1 Available under license

**1.284 kotlinx-coroutines-jdk 1.6.1**

1.284.1 Available under license

**1.285 openapi-generator-core 5.2.1**

1.285.1 Available under license

**1.286 android-device-provider-gradle-proto 30.3.0**

1.286.1 Available under license

**1.287 kotlin-project-model 1.5.31**

1.287.1 Available under license

**1.288 dagger-spi 2.44**

1.288.1 Available under license

**1.289 kotlin-scripting-common 1.5.31**

1.289.1 Available under license

**1.290 dokka-gradle-plugin 1.5.30**

1.290.1 Available under license

**1.291 baselibrary 7.3.0**

1.291.1 Available under license

**1.292 kotlin-gradle-plugin 1.5.31**
1.292.1 Available under license
**1.293 kotlin-compiler-runner 1.5.31**
1.293.1 Available under license
**1.294 dvlib 30.3.0**
1.294.1 Available under license
**1.295 annotation-jvm 1.6.0**
1.295.1 Available under license
**1.296 jetifier-core 1.0.0-beta10**
1.296.1 Available under license
**1.297 collection 1.1.0**
1.297.1 Available under license
**1.298 kotlin-stdlib-common 1.7.10**
1.298.1 Available under license
**1.299 core-proto 0.0.8-alpha07**
1.299.1 Available under license
**1.300 kotlin-reflect 1.7.10**
1.300.1 Available under license
**1.301 kotlin-stdlib 1.5.31**
1.301.1 Available under license
**1.302 javapoet 1.13.0**
1.302.1 Available under license
**1.303 kotlin-daemon-client 1.5.31**
1.303.1 Available under license
**1.304 dagger 2.28.3**
1.304.1 Available under license
**1.305 hilt-compiler 2.44**
1.305.1 Available under license
**1.306 rxkotlin 2.2.0**
1.306.1 Available under license
**1.307 shared 30.3.0**
1.307.1 Available under license
**1.308 kotlin-util-klib 1.5.31**
1.308.1 Available under license
**1.309 jackson-databind 2.12.4**
1.309.1 Available under license
**1.310 kotlin-stdlib-jdk8 1.6.10**
1.310.1 Available under license
**1.311 ddmlib 30.3.0**

1.311.1 Available under license
1.312 kotlin-util-klib 1.7.10
1.312.1 Available under license
1.313 kotlin-daemon-embeddable 1.7.10
1.313.1 Available under license
1.314 kotlin-scripting-common 1.7.10
1.314.1 Available under license
1.315 kotlin-android-extensions-runtime 1.7.10
1.315.1 Available under license
1.316 kotlin-scripting-jvm 1.7.10
1.316.1 Available under license
1.317 kotlin-gradle-plugin-api 1.7.10
1.317.1 Available under license
1.318 markdown-jvm 0.2.4
1.318.1 Available under license
1.319 android-device-provider-ddmlib-proto 30.3.0
1.319.1 Available under license
1.320 core-common 2.1.0
1.320.1 Available under license

# 1.1 trove4j 1.0.20181211

## 1.1.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some

specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who
decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library,
whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that
there is no warranty for the free library. Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.

Finally, software
patents pose a constant threat to the existence of
any free program. We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder. Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The

former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to

exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the
ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete
corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work

that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute
the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany
the work with the complete corresponding
machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood

that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined
with any other library
facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law
if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do
not
excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to

refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our

decision
will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED
ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most
effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

# 1.2 automation 1.11-8

# 1.3 kxml 2.3.0

## 1.3.1 Available under license :

No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2002,2003,2004 Stefan Haustein, Oberhausen, Rhld., Germany
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, sublicense, and/or
* sell copies of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
*

* The  above copyright notice and this permission notice shall be included in
* all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
* AUTHORS OR COPYRIGHT HOLDERS
 BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
* FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
* IN THE SOFTWARE. */
/** Returns the token ID for start tags or the event type for wap proprietary events
   * such as OPAQUE.
   */

Found in path(s):
* /opt/cola/permits/1000022004_1651084632.97/0/kxml2-2-3-0-sources-jar/org/kxml2/wap/WbxmlParser.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2002,2003, Stefan Haustein, Oberhausen, Rhld., Germany
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, sublicense, and/or
* sell copies of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
*
* The  above copyright notice and this permission notice shall be included in
* all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
* AUTHORS OR COPYRIGHT HOLDERS
 BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
* FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
* IN THE SOFTWARE. */

Found in path(s):
* /opt/cola/permits/1000022004_1651084632.97/0/kxml2-2-3-0-sources-jar/org/kxml2/kdom/Node.java
* /opt/cola/permits/1000022004_1651084632.97/0/kxml2-2-3-0-sources-jar/org/kxml2/io/KXmlSerializer.java
* /opt/cola/permits/1000022004_1651084632.97/0/kxml2-2-3-0-sources-jar/org/kxml2/wap/WbxmlSerializer.java
* /opt/cola/permits/1000022004_1651084632.97/0/kxml2-2-3-0-sources-jar/org/kxml2/kdom/Document.java
* /opt/cola/permits/1000022004_1651084632.97/0/kxml2-2-3-0-sources-jar/org/kxml2/wap/Wbxml.java
* /opt/cola/permits/1000022004_1651084632.97/0/kxml2-2-3-0-sources-jar/org/kxml2/io/KXmlParser.java
* /opt/cola/permits/1000022004_1651084632.97/0/kxml2-2-3-0-sources-jar/org/kxml2/kdom/Element.java

# 1.4 juniversalchardet 1.0.3

## 1.4.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.)
You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients
all the rights that
you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under

the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a

special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate
your rights under this License.
However, parties who have received copies, or rights, from you under

this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that

system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author

to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH

YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option)
any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This
is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than 'show w' and 'show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General
Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

# 1.5 txw2 2.3.2

## 1.5.1 Available under license :

Copyright (c) 2018 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF

LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
# Notices for Eclipse Implementation of JAXB

This content is produced and maintained by the Eclipse Implementation of JAXB project.

* Project home: https://projects.eclipse.org/projects/ee4j.jaxb-impl

## Trademarks

Eclipse Implementation of JAXB is a trademark of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Distribution License v. 1.0 which is available at http://www.eclipse.org/org/documents/edl-v10.php.

SPDX-License-Identifier: BSD-3-Clause

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jaxb-ri
* https://github.com/eclipse-ee4j/jaxb-istack-commons
* https://github.com/eclipse-ee4j/jaxb-dtd-parser
* https://github.com/eclipse-ee4j/jaxb-fi
*
 https://github.com/eclipse-ee4j/jaxb-stax-ex
* https://github.com/eclipse-ee4j/jax-rpc-ri

## Third-party Content

This project leverages the following third party content.

Apache Ant (1.10.2)

* License: Apache-2.0 AND W3C AND LicenseRef-Public-Domain

Apache Ant (1.10.2)

* License: Apache-2.0 AND W3C AND LicenseRef-Public-Domain

Apache Felix (1.2.0)

* License: Apache License, 2.0

args4j (2.33)

* License: MIT License

dom4j (1.6.1)

* License: Custom license based on Apache 1.1

file-management (3.0.0)

* License: Apache-2.0
* Project: https://maven.apache.org/shared/file-management/
* Source:
  https://svn.apache.org/viewvc/maven/shared/tags/file-management-3.0.0/

JUnit (4.12)

* License: Eclipse Public License

JUnit (4.12)

* License: Eclipse Public License

maven-compat (3.5.2)

* License: Apache-2.0
* Project: https://maven.apache.org/ref/3.5.2/maven-compat/
* Source:
  https://mvnrepository.com/artifact/org.apache.maven/maven-compat/3.5.2

maven-core
(3.5.2)

* License: Apache-2.0
* Project: https://maven.apache.org/ref/3.5.2/maven-core/index.html
* Source: https://mvnrepository.com/artifact/org.apache.maven/maven-core/3.5.2

maven-plugin-annotations (3.5)

* License: Apache-2.0

* Project: https://maven.apache.org/plugin-tools/maven-plugin-annotations/
* Source:
  https://github.com/apache/maven-plugin-tools/tree/master/maven-plugin-annotations

maven-plugin-api (3.5.2)

* License: Apache-2.0

maven-resolver-api (1.1.1)

* License: Apache-2.0

maven-resolver-api (1.1.1)

* License: Apache-2.0

maven-resolver-connector-basic (1.1.1)

* License: Apache-2.0

maven-resolver-impl (1.1.1)

* License: Apache-2.0

maven-resolver-spi (1.1.1)

* License: Apache-2.0

maven-resolver-transport-file (1.1.1)

* License: Apache-2.0
* Project: https://maven.apache.org/resolver/maven-resolver-transport-file/
* Source:
  https://github.com/apache/maven-resolver/tree/master/maven-resolver-transport-file

maven-resolver-util (1.1.1)

* License: Apache-2.0

maven-settings
 (3.5.2)

* License: Apache-2.0
* Source:
  https://mvnrepository.com/artifact/org.apache.maven/maven-settings/3.5.2

OSGi Service Platform Core Companion Code (6.0)

* License: Apache License, 2.0

plexus-archiver (3.5)

* License: Apache-2.0
* Project: https://codehaus-plexus.github.io/plexus-archiver/
* Source: https://github.com/codehaus-plexus/plexus-archiver

plexus-io (3.0.0)

* License: Apache-2.0

plexus-utils (3.1.0)

* License: Apache- 2.0 or Apache- 1.1 or BSD or Public Domain or Indiana
  University Extreme! Lab Software License V1.1.1 (Apache 1.1 style)

relaxng-datatype (1.0)

* License: New BSD license

Sax (0.2)

* License: SAX-PD
* Project: http://www.megginson.com/downloads/SAX/
* Source: http://sourceforge.net/project/showfiles.php?group_id=29449

testng (6.14.2)

* License: Apache-2.0 AND (MIT OR GPL-1.0+)
* Project: https://testng.org/doc/index.html
* Source: https://github.com/cbeust/testng

wagon-http-lightweight (3.0.0)

* License: Pending
*
 Project: https://maven.apache.org/wagon/
* Source:
  https://mvnrepository.com/artifact/org.apache.maven.wagon/wagon-http-lightweight/3.0.0

xz for java (1.8)

* License: LicenseRef-Public-Domain

## Cryptography

Content may contain encryption software. The country in which you are currently

may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

# 1.6 jakarta xml bind api 2.3.2

## 1.6.1 Available under license :

# Notices for Eclipse Project for JAXB

This content is produced and maintained by the Eclipse Project for JAXB project.

* Project home: https://projects.eclipse.org/projects/ee4j.jaxb

## Trademarks

Eclipse Project for JAXB is a trademark of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Distribution License v. 1.0 which is available at http://www.eclipse.org/org/documents/edl-v10.php.

SPDX-License-Identifier: BSD-3-Clause

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jaxb-api

## Third-party Content

This project leverages the following third party content.

None

## Cryptography

Content may contain encryption software.
The country in which you are currently
may have restrictions on the import, possession, and use, and/or re-export to
another country, of encryption software. BEFORE using any encryption software,
please check the country's laws, regulations and policies concerning the import,
possession, or use, and re-export of encryption software, to see if this is
permitted.

Copyright (c) 2017, 2018 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

- Neither the name of the Eclipse Foundation, Inc. nor the names of its
  contributors may be used to endorse or promote products derived
  from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,

THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1.7 jakarta activation api 1.2.1

## 1.7.1 Available under license :

Copyright (c) 2018 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

- Neither the name of the Eclipse Foundation, Inc. nor the names of its
  contributors may be used to endorse or promote products derived
  from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Notices for Eclipse Project for JAF

This content is produced and maintained by the Eclipse Project for JAF project.

* Project home: https://projects.eclipse.org/projects/ee4j.jaf

## Copyright

All content is the property of the respective authors or their employers. For
more information regarding authorship of content, please consult the listed
source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms
of the Eclipse Distribution License v. 1.0,
which is available at http://www.eclipse.org/org/documents/edl-v10.php.

SPDX-License-Identifier: BSD-3-Clause

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jaf

## Third-party Content

This project leverages the following third party content.

JUnit (4.12)

* License: Eclipse Public License
Eclipse Distribution License - v 1.0
Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
/*
* Copyright (c) 1997, 2018 Oracle and/or its affiliates. All rights reserved.
*
* This program and the accompanying materials are made available under the
* terms of the Eclipse Distribution License v. 1.0, which is available at
* http://www.eclipse.org/org/documents/edl-v10.php.
*
* SPDX-License-Identifier: BSD-3-Clause
*/

# 1.8 generex 1.0.2

## 1.8.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 y.mifrah
*

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1000610466_1606952086.17/0/generex-1-0-2-sources-1-
jar/com/mifmif/common/regex/GenerexIterator.java
* /opt/cola/permits/1000610466_1606952086.17/0/generex-1-0-2-sources-1-
jar/com/mifmif/common/regex/Generex.java
* /opt/cola/permits/1000610466_1606952086.17/0/generex-1-0-2-sources-1-
jar/com/mifmif/common/regex/Node.java
*
 /opt/cola/permits/1000610466_1606952086.17/0/generex-1-0-2-sources-1-
jar/com/mifmif/common/regex/util/Iterator.java
* /opt/cola/permits/1000610466_1606952086.17/0/generex-1-0-2-sources-1-
jar/com/mifmif/common/regex/Main.java
* /opt/cola/permits/1000610466_1606952086.17/0/generex-1-0-2-sources-1-
jar/com/mifmif/common/regex/util/Iterable.java
```

No license file was found, but licenses were detected in source scan.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
 <modelVersion>4.0.0</modelVersion>
 <groupId>com.github.mifmif</groupId>
 <artifactId>generex</artifactId>
 <version>1.0.2</version>
 <name>Generex</name>
 <url>https://github.com/mifmif/Generex/tree/master</url>
 <description>Generex A Java Library for regex to Strings generation</description>
 <packaging>jar</packaging>
 <licenses>
```

```
 <license>
  <name>The Apache Software License, Version 2.0</name>
  <url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
  <distribution>repo</distribution>
 </license>
</licenses>
<developers>
 <developer>
  <id>mifmif</id>
  <name>MIFRAH Youssef</name>
  <email>mifmif.com@gmail.com</email>
 </developer>
 <developer>
  <id>mkolisnyk</id>
  <name>mkolisnyk</name>
  <email>kolesnik.nickolay@gmail.com</email>
 </developer>
</developers>

<parent>
 <groupId>org.sonatype.oss</groupId>
 <artifactId>oss-parent</artifactId>
 <version>7</version>
</parent>

<properties>
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
 <maven.pmd.excludes>**/generated-sources/**</maven.pmd.excludes>
</properties>
<scm>
 <connection>scm:git:git@github.com:mifmif/Generex.git</connection>
 <developerConnection>scm:git:git@github.com:mifmif/Generex.git</developerConnection>
 <url>https://github.com/mifmif/Generex.git</url>
</scm>
<ciManagement>
 <url>https://travis-ci.org/mifmif/Generex/builds</url>
 <system>Travis</system>
 <notifiers>
  <notifier>
   <address>mifmif.com@gmail.com</address>
   <type>mail</type>
  </notifier>
  <notifier>
   <address>kolesnik.nickolay@gmail.com</address>
   <type>mail</type>
  </notifier>
 </notifiers>
</ciManagement>
```

```
<distributionManagement>
 <snapshotRepository>
  <id>ossrh</id>
  <url>https://oss.sonatype.org/content/repositories/snapshots</url>
 </snapshotRepository>
 <repository>
  <id>ossrh</id>
  <url>https://oss.sonatype.org/service/local/staging/deploy/maven2/</url>
 </repository>
</distributionManagement>


<build>
 <sourceDirectory>src/main/java</sourceDirectory>
 <testSourceDirectory>src/test/java</testSourceDirectory>
 <resources>
  <resource>
   <directory>src</directory>
   <excludes>
    <exclude>**/*.java</exclude>
   </excludes>
  </resource>
  <resource>
   <directory>target/dependency</directory>
   <excludes>
    <exclude>**/*.java</exclude>
   </excludes>
  </resource>
  <resource>
   <directory>src/main/resources</directory>
   <includes>
    <include>**/*.properties</include>
   </includes>
  </resource>
 </resources>
 <plugins>
  <plugin>
   <groupId>org.apache.maven.plugins</groupId>
   <artifactId>maven-install-plugin</artifactId>
   <version>2.5.2</version>
   <configuration>
    <file>target/${project.artifactId}-${project.version}.jar</file>
    <groupId>${project.groupId}</groupId>
    <artifactId>${project.artifactId}</artifactId>
    <version>${project.version}</version>
    <packaging>maven-plugin</packaging>
   </configuration>
```

```
</plugin>

<plugin>
 <artifactId>maven-compiler-plugin</artifactId>
 <version>3.5.1</version>
 <configuration>
  <source>1.5</source>
  <target>1.5</target>
 </configuration>
</plugin>
<plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-jar-plugin</artifactId>
 <version>3.0.2</version>
 <configuration>
  <excludes>
   <exclude>*</exclude>
   <exclude>com/thoughtworks/**/*</exclude>
   <exclude>freemarker/**/*</exclude>
   <exclude>ftl/**/*</exclude>
   <exclude>i18n/**/*</exclude>
   <exclude>style/**/*</exclude>
   <exclude>junit/**/*</exclude>
   <exclude>licenses/**/*</exclude>
   <exclude>META-INF/maven/**/*</exclude>
   <exclude>org/codehaus/**/*</exclude>
   <exclude>org/hamcrest/**/*</exclude>
   <exclude>org/jbehave/**/*</exclude>
   <exclude>org/junit/**/*</exclude>
   <exclude>org/testng/**/*</exclude>
   <exclude>org/xmlpull/**/*</exclude>
   <exclude>stories/**/*</exclude>
   <exclude>style/**/*</exclude>
   <exclude>tests/**/*</exclude>
  </excludes>
  <archive>
   <manifest>
    <addClasspath>false</addClasspath>
    <addDefaultImplementationEntries>true</addDefaultImplementationEntries>
    <addDefaultSpecificationEntries>true</addDefaultSpecificationEntries>
    <addExtensions>false</addExtensions>
    <classpathLayoutType>simple</classpathLayoutType>
   </manifest>
  </archive>
 </configuration>
</plugin>
<plugin>
 <artifactId>maven-assembly-plugin</artifactId>
```

```
  <version>2.6</version>
  <configuration>
   <descriptors>
    <descriptor>./sources.xml</descriptor>
   </descriptors>
  </configuration>
 </plugin>
 <plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-dependency-plugin</artifactId>
  <version>2.10</version>
 </plugin>
 <plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-javadoc-plugin</artifactId>
  <version>2.10.3</version>
  <configuration>
   <show>private</show>
   <nohelp>false</nohelp>
   <failOnError>false</failOnError>
   <sourcepath>src/main/java</sourcepath>
  </configuration>
 </plugin>
 <plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-release-plugin</artifactId>
  <configuration>
   <tagNameFormat>v@{project.version}</tagNameFormat>
   <preparationGoals>package
install:install-file</preparationGoals>
   <scmCommentPrefix>#3</scmCommentPrefix>
  </configuration>
 </plugin>
 <plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-deploy-plugin</artifactId>
  <version>2.8.2</version>
  <configuration>
   <skip>true</skip>
  </configuration>
 </plugin>


 <plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-source-plugin</artifactId>
  <version>3.0.1</version>
  <executions>
```

```
  <execution>
   <goals>
    <goal>jar-no-fork</goal>
   </goals>
  </execution>
 </executions>
</plugin>


<plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-javadoc-plugin</artifactId>
 <executions>
  <execution>
   <id>attach-javadocs</id>
   <goals>
    <goal>jar</goal>
   </goals>
  </execution>
 </executions>
</plugin>


<plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-gpg-plugin</artifactId>
 <executions>
  <execution>
   <id>sign-artifacts</id>
   <phase>deploy</phase>
   <goals>
    <goal>sign</goal>
   </goals>
  </execution>
 </executions>


</plugin>


<plugin>
 <groupId>org.sonatype.plugins</groupId>
 <artifactId>nexus-staging-maven-plugin</artifactId>
 <version>1.6.3</version>
 <executions>
  <execution>
   <id>default-deploy</id>
   <phase>deploy</phase>
```

```
   <goals>
    <goal>deploy</goal>
   </goals>
  </execution>
 </executions>
 <extensions>true</extensions>
 <configuration>
  <serverId>ossrh</serverId>
  <nexusUrl>https://oss.sonatype.org/</nexusUrl>
 </configuration>
</plugin>
<plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-pmd-plugin</artifactId>
 <version>3.6</version>
 <configuration>
  <failOnViolation>true</failOnViolation>
  <excludeRoots>
   <excludeRoot>target/generated-sources/plugin</excludeRoot>
  </excludeRoots>
  <excludes>
   <exclude>**/HelpMojo.java</exclude>
  </excludes>
 </configuration>
 <executions>
  <execution>
   <goals>
    <goal>check</goal>
    <goal>cpd-check</goal>
   </goals>
   <configuration>
    <failOnViolation>true</failOnViolation>
    <excludes>
     <exclude>**/HelpMojo.java</exclude>
    </excludes>
   </configuration>
  </execution>
 </executions>
</plugin>
<plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-checkstyle-plugin</artifactId>
 <version>2.10</version>
 <configuration>
  <excludes>**/test/java/**/*.java</excludes>
  <configLocation>./sun_checks.xml</configLocation>
  <suppressionsLocation>./checkstyle-suppressions.xml</suppressionsLocation>
  <failOnViolation>true</failOnViolation>
```

```
 </configuration>
 <executions>
  <execution>
   <goals>
    <goal>check</goal>
   </goals>
  </execution>
 </executions>
</plugin>
<plugin>
 <groupId>org.codehaus.mojo</groupId>
 <artifactId>cobertura-maven-plugin</artifactId>
 <version>2.6</version>
 <configuration>
  <check>
   <haltOnFailure>false</haltOnFailure>
   <totalBranchRate>80</totalBranchRate>
   <totalLineRate>80</totalLineRate>
   <packageLineRate>80</packageLineRate>
   <packageBranchRate>80</packageBranchRate>
  </check>
  <instrumentation>
   <excludes>
    <exclude>**/HelpMojo.class</exclude>
   </excludes>
  </instrumentation>
 </configuration>
 <executions>
  <execution>
   <id>verification</id>
   <phase>verify</phase>
   <goals>
    <goal>check</goal>
    <goal>cobertura</goal>
   </goals>
  </execution>
 </executions>
</plugin>
<plugin>
 <groupId>org.codehaus.mojo</groupId>
 <artifactId>javancss-maven-plugin</artifactId>
 <version>2.0</version>
 <configuration>
  <encoding>${project.build.sourceEncoding}</encoding>
  <ccnLimit>15</ccnLimit>
  <ncssLimit>100</ncssLimit>
  <failOnViolation>true</failOnViolation>
  <excludes>
```

```
    <exclude>**/tests/**/*.*</exclude>
   </excludes>
  </configuration>
  <executions>
   <execution>
    <goals>
     <goal>check</goal>
    </goals>
   </execution>
  </executions>
 </plugin>
 <plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>findbugs-maven-plugin</artifactId>
  <version>3.0.1</version>
  <configuration>
   <encoding>${project.build.sourceEncoding}</encoding>
   <failOnError>true</failOnError>
   <excludeFilterFile>findBugsExclude.xml</excludeFilterFile>
  </configuration>
  <executions>
   <execution>
    <goals>
     <goal>check</goal>
    </goals>
   </execution>
  </executions>
 </plugin>
 <plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>jdepend-maven-plugin</artifactId>
  <version>2.0</version>
 </plugin>
</plugins>
</build>
<dependencies>
 <dependency>
  <groupId>dk.brics.automaton</groupId>
  <artifactId>automaton</artifactId>
  <version>1.11-8</version>
 </dependency>
 <dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.12</version>
  <scope>test</scope>
 </dependency>
 <dependency>
```

```
  <groupId>org.hamcrest</groupId>
  <artifactId>hamcrest-junit</artifactId>
  <version>2.0.0.0</version>
  <scope>test</scope>
 </dependency>
</dependencies>
</project>
```

Found
in path(s):
* /opt/cola/permits/1000610466_1606952086.17/0/generex-1-0-2-sources-1-jar/META-INF/maven/com.github.mifmif/generex/pom.xml

# 1.9 commons-logging 1.2

## 1.9.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Commons Logging
Copyright 2003-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

# 1.10 commons-lang3 3.7

## 1.10.1 Available under license :

Apache License
Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or
(iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from)
the Work and for which the
editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication
on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant
of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as
of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own
copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names

of the Licensor,
except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law
(such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your
sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Commons Lang
Copyright 2001-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: StringUtils.containsWhitespace())

# 1.11 gson 2.8.0

## 1.11.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and

limitations under the License.

# 1.12 json-simple 1.1

## 1.12.1 Available under license :

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source"
form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely
link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright

owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean
Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute,
all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the
NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions

stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the
trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor
harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: HOW TO APPLY THE APACHE LICENSE TO YOUR WORK
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless
required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.13 antlr-runtime 4.5.3

## 1.13.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
*  Copyright (c) 2014 Sam Harwell
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
*  OF MERCHANTABILITY
 AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/dfa/EmptyEdgeMap.java

No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
*  Copyright (c) 2014 Terence Parr
*  Copyright (c) 2014 Sam Harwell
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 THE IMPLIED WARRANTIES
*  OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/dfa/AcceptStateInfo.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ErrorInfo.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/DecisionEventInfo.java
*
 /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/VocabularyImpl.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ContextSensitivityInfo.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/DecisionInfo.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-

jar/org/antlr/v4/runtime/atn/PredicateEvalInfo.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LookaheadEventInfo.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ConflictInfo.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ParseInfo.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/AmbiguityInfo.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/Vocabulary.java
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2012 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/Func1.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-

jar/org/antlr/v4/runtime/misc/Args.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-
jar/org/antlr/v4/runtime/misc/Predicate.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-
jar/org/antlr/v4/runtime/ProxyParserErrorListener.java
*
/opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-
jar/org/antlr/v4/runtime/RuleVersion.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-
jar/org/antlr/v4/runtime/misc/Func0.java
No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
*  Copyright (c) 2012 Sam Harwell
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
*  OF MERCHANTABILITY
 AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-
jar/org/antlr/v4/runtime/dfa/EdgeMap.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-
jar/org/antlr/v4/runtime/atn/SimulatorState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-
jar/org/antlr/v4/runtime/dfa/AbstractEdgeMap.java

* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/dfa/SingletonEdgeMap.java
*
 /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/dfa/SparseEdgeMap.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/dfa/ArrayEdgeMap.java
No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
*  Copyright (c) 2012 Terence Parr
*  Copyright (c) 2012 Sam Harwell
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 THE IMPLIED WARRANTIES
*  OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/SyntaxTree.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/TokenFactory.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/TestRig.java
*

/opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/MultiMap.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/ParserRuleContext.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/ParseTree.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/Lexer.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/FlexibleHashMap.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/ParseTreeProperty.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/AtomTransition.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/StarBlockStartState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ParserATNSimulator.java
*
/opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/SingletonPredictionContext.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/Dependents.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/EqualityComparator.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/ANTLRFileStream.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/ParseTreeWalker.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/LexerInterpreter.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ATNState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/TokenStreamRewriter.java
*
/opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/CharStream.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/LogManager.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/RangeTransition.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/RuleTransition.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ArrayPredictionContext.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/ProxyErrorListener.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-

jar/org/antlr/v4/runtime/misc/MurmurHash.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/PredicateTransition.java
*
 /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/CommonToken.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/RuleStopState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/ErrorNode.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/dfa/LexerDFASerializer.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/Array2DHashSet.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/BlockEndState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/BufferedTokenStream.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/PrecedencePredicateTransition.java
*
 /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/InputMismatchException.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/WildcardTransition.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/RuleContext.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/Interval.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/Tree.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/ANTLRErrorStrategy.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/NotSetTransition.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/TokensStartState.java
*
 /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/OrderedATNConfigSet.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ATN.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/Utils.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/LexerNoViableAltException.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/BasicState.java

* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/UnbufferedTokenStream.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ATNConfig.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ATNSerializer.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/IntegerList.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/PredictionContextCache.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/AbstractPredicateTransition.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/NoViableAltException.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/PredictionContext.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/Transition.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/WritableToken.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/DefaultErrorStrategy.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/IntegerStack.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/PlusBlockStartState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/ANTLRInputStream.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/CommonTokenStream.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/SemanticContext.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/RuleStartState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/TerminalNode.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/dfa/DFA.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/BasicBlockStartState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ActionTransition.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/Parser.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/BlockStartState.java

* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/EpsilonTransition.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/PlusLoopbackState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/RuleDependency.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/StarLoopbackState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/ParseTreeVisitor.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/UnbufferedCharStream.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/TerminalNodeImpl.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/DecisionState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/RuleDependencies.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/ParseTreeListener.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/DoubleKeyMap.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/IntStream.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/ANTLRErrorListener.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/CommonTokenFactory.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/RuleNode.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/ObjectEqualityComparator.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/EmptyPredictionContext.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/AbstractParseTreeVisitor.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LexerATNSimulator.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/TokenStream.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/TokenSource.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/IntSet.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-

jar/org/antlr/v4/runtime/atn/LoopEndState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/AbstractEqualityComparator.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/IntervalSet.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/ParseCancellationException.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/ErrorNodeImpl.java
*
/opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ATNConfigSet.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/Recognizer.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/DiagnosticErrorListener.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/Trees.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/dfa/DFASerializer.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/StarLoopEntryState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/PredictionMode.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/OrderedHashSet.java
*
/opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/BaseErrorListener.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/FailedPredicateException.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/dfa/DFAState.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/Token.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/BailErrorStrategy.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/RecognitionException.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LL1Analyzer.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/ConsoleErrorListener.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/SetTransition.java
No license file was found, but licenses were detected in source scan.

/*
* [The "BSD license"]

* Copyright (c) 2013 Terence Parr
* Copyright (c) 2013 Sam Harwell
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*  notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*  notice, this list of conditions and the following disclaimer in the
*  documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*  derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LexerIndexedCustomAction.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LexerTypeAction.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ATNType.java
*
 /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ATNDeserializationOptions.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LexerAction.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ATNSimulator.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LexerPushModeAction.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ATNDeserializer.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-

jar/org/antlr/v4/runtime/atn/LexerPopModeAction.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LexerModeAction.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/ProfilingATNSimulator.java
*
/opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LexerSkipAction.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LexerChannelAction.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LexerMoreAction.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LexerActionType.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LexerActionExecutor.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/atn/LexerCustomAction.java
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2012 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/Tuple3.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/Tuple2.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/Tuple.java
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2011 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/ParserErrorListener.java
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]

Copyright (c) 2012 Terence Parr
Copyright (c) 2012 Sam Harwell
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS
 FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/RuleDependencyProcessor.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/misc/RuleDependencyChecker.java
No license file was found, but licenses were detected in source scan.

/*
* [The "BSD license"]
* Copyright (c) 2013 Terence Parr
* Copyright (c) 2013 Sam Harwell
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/pattern/ParseTreePatternMatcher.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/pattern/TextChunk.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/InterpreterRuleContext.java
*
 /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/pattern/TokenTagToken.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/pattern/Chunk.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/pattern/TagChunk.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/pattern/ParseTreeMatch.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/pattern/RuleTagToken.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/ParserInterpreter.java
* /opt/cola/permits/1000876358_1646171538.93/0/antlr4-runtime-4-5-3-sources-jar/org/antlr/v4/runtime/tree/pattern/ParseTreePattern.java

# 1.14 j2objc-annotations 1.3

## 1.14.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1000890861_1651084689.86/0/j2objc-annotations-1-3-sources-jar/com/google/j2objc/annotations/RetainedWith.java
* /opt/cola/permits/1000890861_1651084689.86/0/j2objc-annotations-1-3-sources-jar/com/google/j2objc/annotations/ReflectionSupport.java
* /opt/cola/permits/1000890861_1651084689.86/0/j2objc-annotations-1-3-sources-jar/com/google/j2objc/annotations/RetainedLocalRef.java
* /opt/cola/permits/1000890861_1651084689.86/0/j2objc-annotations-1-3-sources-jar/com/google/j2objc/annotations/Property.java
* /opt/cola/permits/1000890861_1651084689.86/0/j2objc-annotations-1-3-sources-jar/com/google/j2objc/annotations/LoopTranslation.java
* /opt/cola/permits/1000890861_1651084689.86/0/j2objc-annotations-1-3-sources-jar/com/google/j2objc/annotations/J2ObjCIncompatible.java
* /opt/cola/permits/1000890861_1651084689.86/0/j2objc-annotations-1-3-sources-jar/com/google/j2objc/annotations/ObjectiveCName.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012 Google Inc. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1000890861_1651084689.86/0/j2objc-annotations-1-3-sources-jar/com/google/j2objc/annotations/Weak.java
* /opt/cola/permits/1000890861_1651084689.86/0/j2objc-annotations-1-3-sources-jar/com/google/j2objc/annotations/AutoreleasePool.java
* /opt/cola/permits/1000890861_1651084689.86/0/j2objc-annotations-1-3-sources-jar/com/google/j2objc/annotations/WeakOuter.java

# 1.15 javax-activation 1.2.0

## 1.15.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software

(or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software

in the absence of
Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it
absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any
terms which differ from this License
are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or

otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License
if You: (a) rename the license and
remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED
HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer
(if the
Initial Developer is not the Participant) and all Contributors under

Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent

with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer

software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of
Goods is expressly excluded. Any law or
regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

---

NOTICE PURSUANT TO SECTION
9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION
LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the

State of California (excluding conflict-of-law provisions). Any
litigation relating to this License shall be subject to the jurisdiction
of the Federal Courts of the Northern District of California and the
state courts of the State of California, with venue lying in Santa Clara
County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to
share and change it. By contrast, the GNU General Public License is
intended to guarantee your freedom to share and change free software--to
make sure the software is free for all its users.
This General Public
License applies to most of the Free Software Foundation's software and
to any other program whose authors commit to using it. (Some other Free
Software Foundation software is covered by the GNU Library General
Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.
Our General Public Licenses are designed to make sure that you have the
freedom to distribute copies of free software (and charge for this
service if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone
to deny you these rights or to ask you to surrender the rights. These
restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute
copies of such a program, whether gratis
or for a fee, you must give the recipients all the rights that you have.
You must make sure that they, too, receive or can get the source code.
And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be
a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License may
add an explicit geographical distribution limitation excluding those
countries, so that distribution is permitted only in or among countries
not thus excluded. In such case, this License incorporates the
limitation as if written in the body of this License.

9. The Free Software Foundation
may publish revised and/or new
versions of the General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Program does not specify a version
number of this License, you may choose any version ever published by the
Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the
author to ask for permission. For software which is copyrighted by the
Free Software Foundation, write to the Free Software Foundation; we
sometimes make exceptions for this. Our decision will be
guided by the
two goals of preserving the free status of all derivatives of our free
software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND,
EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH
YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL
NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY

AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL
DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM
(INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED
INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF
THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR
OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to
attach them to the start of each source file to most effectively convey
the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief
idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version
69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type

`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by
James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

#

Certain source files distributed by Oracle America, Inc. and/or its affiliates are subject to the following clarification and special exception to the GPLv2, based on the GNU Project exception for its Classpath libraries, known as the GNU Classpath Exception, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the LICENSE file that accompanied this code."

You should also note that Oracle includes multiple, independent programs in this software package. Some of those programs are provided under licenses deemed incompatible with the GPLv2 by the Free Software Foundation and others. For example, the package includes programs licensed under the Apache License, Version 2.0. Such programs are licensed to you under their original licenses.

Oracle facilitates your further distribution of this package by adding the Classpath Exception to the necessary parts of its GPLv2 code, which permits you to use that code in combination with other independent

modules not licensed under the GPLv2. However, note that this would not permit you to commingle code under an incompatible license with Oracle's GPLv2 licensed code by, for example, cutting and pasting such code into a file also containing Oracle's GPLv2 licensed code and then distributing the result. Additionally, if you were to remove the Classpath Exception from any of the files to which it applies and distribute the result, you would likely be required to license some or all of the other code in that distribution under the GPLv2 as well, and since the GPLv2 is incompatible with the license terms of some items included
in the distribution by Oracle, removing the Classpath
Exception could therefore effectively compromise your ability to further distribute the package.

Proceed with caution and we recommend that you obtain the advice of a lawyer skilled in open source matters before removing the Classpath Exception or making modifications to this package which may subsequently be redistributed and/or involve the use of third party software.

CLASSPATH EXCEPTION
Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet,
for each linked
independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.
<html>
<head>
<title>Specification License</title>
</head>
<body>
Specification: JSR-925 JavaBeans Activation Framework ("Specification")
<br/>
Version: 1.2
<br/>
Status: Maintenance Release

<br/>
Specification Lead: Oracle America, Inc. ("Specification Lead")
<br/>
Release: July 2017
<br/>

<br/>
Copyright © 2017 Oracle America, Inc.  All rights reserved.
<br/>

LIMITED LICENSE GRANTS
<br/>
1. License for Evaluation Purposes. Specification Lead hereby grants you a fully-paid, non-exclusive, non-transferable, worldwide, limited license (without the right to sublicense), under Specification Lead's applicable intellectual property rights to view, download, use and reproduce the Specification only for the purpose of internal evaluation.  This includes (i) developing applications intended to run on an implementation of the Specification, provided that such applications do not themselves implement any portion(s) of the Specification, and (ii) discussing the Specification with any third party;
 and (iii) excerpting brief portions of the Specification in oral or written communications which discuss the Specification provided that such excerpts do not in the aggregate constitute a significant portion of the Specification.

2. License for the Distribution of Compliant Implementations. Specification Lead  also grants you a perpetual, non-exclusive, non-transferable, worldwide, fully paid-up, royalty free, limited license (without the right to sublicense) under any applicable copyrights or, subject to the provisions of subsection 4 below, patent rights it may have covering the Specification to create and/or distribute an Independent Implementation of the Specification that: (a) fully implements the Specification including all its required interfaces and functionality; (b) does not modify, subset, superset or otherwise extend the Licensor Name Space, or include any public or protected packages, classes, Java interfaces, fields or methods within the Licensor Name Space
 other than those required/authorized by the
Specification or Specifications being implemented; and (c) passes the Technology Compatibility Kit (including satisfying the requirements of the applicable TCK Users Guide) for such Specification ("Compliant Implementation").  In addition, the foregoing license is expressly conditioned on your not acting outside its scope.  No license is granted hereunder for any other purpose (including, for example, modifying the Specification, other than to the extent of your fair use

rights, or distributing the Specification to third parties). Also, no right, title, or interest in or to any trademarks, service marks, or trade names of Specification Lead or Specification Lead's licensors is granted hereunder. Java, and Java-related logos, marks and names are trademarks or registered trademarks of Oracle America, Inc. in the U.S. and other countries.

3. Pass-through Conditions. You need not include limitations (a)-(c) from the previous paragraph
or any other particular "pass through"
requirements in any license You grant concerning the use of your Independent Implementation or products derived from it. However, except with respect to Independent Implementations (and products derived from them) that satisfy limitations (a)-(c) from the previous paragraph, You may neither: (a) grant or otherwise pass through to your licensees any licenses under Specification Lead's applicable intellectual property rights; nor (b) authorize your licensees to make any claims concerning their implementation's compliance with the Specification in question.

4. Reciprocity Concerning Patent Licenses.
<br/>
a. With respect to any patent claims covered by the license granted under subparagraph 2 above that would be infringed by all technically feasible implementations of the Specification, such license is conditioned upon your offering on fair, reasonable and non-discriminatory terms, to any party seeking it from You, a perpetual, non-exclusive,
non-transferable, worldwide license under
Your patent rights which are or would be infringed by all technically feasible implementations of the Specification to develop, distribute and use a Compliant Implementation.
<br/>
b. With respect to any patent claims owned by Specification Lead and covered by the license granted under subparagraph 2, whether or not their infringement can be avoided in a technically feasible manner when implementing the Specification, such license shall terminate with respect to such claims if You initiate a claim against Specification Lead that it has, in the course of performing its responsibilities as the Specification Lead, induced any other entity to infringe Your patent rights.
<br/>
c. Also with respect to any patent claims owned by Specification Lead and covered by the license granted under subparagraph 2 above, where the infringement of such claims can be avoided in a technically feasible manner when implementing the Specification such license,

with
respect to such claims, shall terminate if You initiate a claim against Specification Lead that its making, having made, using, offering to sell, selling or importing a Compliant Implementation infringes Your patent rights.

5. Definitions. For the purposes of this Agreement: "Independent Implementation" shall mean an implementation of the Specification that neither derives from any of Specification Lead's source code or binary code materials nor, except with an appropriate and separate license from Specification Lead, includes any of Specification Lead's source code or binary code materials; "Licensor Name Space" shall mean the public class or interface declarations whose names begin with "java", "javax", "com.oracle", "com.sun" or their equivalents in any subsequent naming convention adopted by Oracle America, Inc. through the Java Community Process, or any recognized successors or replacements thereof; and "Technology Compatibility Kit" or "TCK" shall mean the test
suite and accompanying TCK User's Guide provided by Specification Lead which corresponds to the Specification and that was available either (i) from Specification Lead 120 days before the first release of Your Independent Implementation that allows its use for commercial purposes, or (ii) more recently than 120 days from such release but against which You elect to test Your implementation of the Specification.

This Agreement will terminate immediately without notice from Specification Lead if you breach the Agreement or act outside the scope of the licenses granted above.

DISCLAIMER OF WARRANTIES
<br/>
THE SPECIFICATION IS PROVIDED "AS IS". SPECIFICATION LEAD MAKES NO REPRESENTATIONS OR WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT (INCLUDING AS A CONSEQUENCE OF ANY PRACTICE OR IMPLEMENTATION OF THE SPECIFICATION), OR THAT THE CONTENTS OF THE SPECIFICATION
ARE SUITABLE FOR ANY PURPOSE. This document does not represent any commitment to release or implement any portion of the Specification in any product. In addition, the Specification could include technical inaccuracies or typographical errors.

LIMITATION OF LIABILITY

<br/>
TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL SPECIFICATION LEAD OR ITS LICENSORS BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUE, PROFITS OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED IN ANY WAY TO YOUR HAVING, IMPLEMENTING OR OTHERWISE USING THE SPECIFICATION, EVEN IF SPECIFICATION LEAD AND/OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. You will indemnify, hold harmless, and defend Specification Lead and its licensors from any claims arising or resulting from: (i) your use of the Specification; (ii) the use or distribution of your
Java application, applet and/or implementation;
and/or (iii) any claims that later versions or releases of any Specification furnished to you are incompatible with the Specification provided to you under this license.

RESTRICTED RIGHTS LEGEND
<br/>
U.S. Government: If this Specification is being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in the Software and accompanying documentation shall be only as set forth in this license; this is in accordance with 48 C.F.R. 227.7201 through 227.7202-4 (for Department of Defense (DoD) acquisitions) and with 48 C.F.R. 2.101 and 12.212 (for non-DoD acquisitions).

REPORT
<br/>
If you provide Specification Lead with any comments or suggestions concerning the Specification ("Feedback"), you hereby: (i) agree that such Feedback is provided on a non-proprietary and non-confidential basis, and (ii) grant Specification Lead a perpetual,
non-exclusive,
worldwide, fully paid-up, irrevocable license, with the right to sublicense through multiple levels of sublicensees, to incorporate, disclose, and use without limitation the Feedback for any purpose.

GENERAL TERMS
<br/>
Any action related to this Agreement will be governed by California law and controlling U.S. federal law. The U.N. Convention for the International Sale of Goods and the choice of law rules of any jurisdiction will not apply.

The Specification is subject to U.S. export control laws and may be subject to export or import regulations in other countries. Licensee agrees to comply strictly with all such laws and regulations and acknowledges that it has the responsibility to obtain such licenses to export, re-export or import as may be required after delivery to Licensee.

This Agreement is the parties' entire agreement relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals,
 conditions, representations and warranties
and prevails over any conflicting or additional terms of any quote, order,  acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification to this Agreement will be binding, unless in writing and signed by an authorized representative of each party.

</body>
</html>

# 1.16 jaxb-runtime 2.3.2

## 1.16.1 Available under license :

Copyright (c) 2018 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Notices for Eclipse Implementation of JAXB

This content is produced and maintained by the Eclipse Implementation of JAXB project.

* Project home: https://projects.eclipse.org/projects/ee4j.jaxb-impl

## Trademarks

Eclipse Implementation of JAXB is a trademark of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Distribution License v. 1.0 which is available at http://www.eclipse.org/org/documents/edl-v10.php.

SPDX-License-Identifier: BSD-3-Clause

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jaxb-ri
* https://github.com/eclipse-ee4j/jaxb-istack-commons
* https://github.com/eclipse-ee4j/jaxb-dtd-parser
* https://github.com/eclipse-ee4j/jaxb-fi
* https://github.com/eclipse-ee4j/jaxb-stax-ex
* https://github.com/eclipse-ee4j/jax-rpc-ri

## Third-party Content

This project leverages the following third party content.

Apache Ant (1.10.2)

* License: Apache-2.0 AND W3C AND LicenseRef-Public-Domain

Apache Ant (1.10.2)

* License: Apache-2.0 AND W3C AND LicenseRef-Public-Domain

Apache Felix (1.2.0)

* License: Apache License, 2.0

args4j (2.33)

* License: MIT License

dom4j (1.6.1)

* License: Custom license based on Apache 1.1

file-management (3.0.0)

* License: Apache-2.0
* Project: https://maven.apache.org/shared/file-management/
* Source:
  https://svn.apache.org/viewvc/maven/shared/tags/file-management-3.0.0/

JUnit (4.12)

* License: Eclipse Public License

JUnit (4.12)

* License: Eclipse Public License

maven-compat (3.5.2)

* License: Apache-2.0
* Project: https://maven.apache.org/ref/3.5.2/maven-compat/
* Source:
  https://mvnrepository.com/artifact/org.apache.maven/maven-compat/3.5.2

maven-core
(3.5.2)

* License: Apache-2.0
* Project: https://maven.apache.org/ref/3.5.2/maven-core/index.html

* Source: https://mvnrepository.com/artifact/org.apache.maven/maven-core/3.5.2

maven-plugin-annotations (3.5)

* License: Apache-2.0
* Project: https://maven.apache.org/plugin-tools/maven-plugin-annotations/
* Source:
  https://github.com/apache/maven-plugin-tools/tree/master/maven-plugin-annotations

maven-plugin-api (3.5.2)

* License: Apache-2.0

maven-resolver-api (1.1.1)

* License: Apache-2.0

maven-resolver-api (1.1.1)

* License: Apache-2.0

maven-resolver-connector-basic (1.1.1)

* License: Apache-2.0

maven-resolver-impl (1.1.1)

* License: Apache-2.0

maven-resolver-spi (1.1.1)

* License: Apache-2.0

maven-resolver-transport-file (1.1.1)

* License: Apache-2.0
* Project: https://maven.apache.org/resolver/maven-resolver-transport-file/
* Source:
  https://github.com/apache/maven-resolver/tree/master/maven-resolver-transport-file

maven-resolver-util (1.1.1)

* License: Apache-2.0

maven-settings
(3.5.2)

* License: Apache-2.0

* Source:
  https://mvnrepository.com/artifact/org.apache.maven/maven-settings/3.5.2

OSGi Service Platform Core Companion Code (6.0)

* License: Apache License, 2.0

plexus-archiver (3.5)

* License: Apache-2.0
* Project: https://codehaus-plexus.github.io/plexus-archiver/
* Source: https://github.com/codehaus-plexus/plexus-archiver

plexus-io (3.0.0)

* License: Apache-2.0

plexus-utils (3.1.0)

* License: Apache- 2.0 or Apache- 1.1 or BSD or Public Domain or Indiana
  University Extreme! Lab Software License V1.1.1 (Apache 1.1 style)

relaxng-datatype (1.0)

* License: New BSD license

Sax (0.2)

* License: SAX-PD
* Project: http://www.megginson.com/downloads/SAX/
* Source: http://sourceforge.net/project/showfiles.php?group_id=29449

testng (6.14.2)

* License: Apache-2.0 AND (MIT OR GPL-1.0+)
* Project: https://testng.org/doc/index.html
* Source: https://github.com/cbeust/testng

wagon-http-lightweight (3.0.0)

* License: Pending
*
 Project: https://maven.apache.org/wagon/
* Source:
  https://mvnrepository.com/artifact/org.apache.maven.wagon/wagon-http-lightweight/3.0.0

xz for java (1.8)

* License: LicenseRef-Public-Domain

## Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

# 1.17 stax-ex 1.8.1

## 1.17.1 Available under license :

Copyright (c) 2017 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Notices for Eclipse Implementation of JAXB

This content is produced and maintained by the Eclipse Implementation of JAXB

project.

* Project home: https://projects.eclipse.org/projects/ee4j.jaxb-impl

## Trademarks

Eclipse Implementation of JAXB is a trademark of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For
more information regarding authorship of content, please consult the listed
source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms
of the Eclipse Distribution License v. 1.0 which is available at
http://www.eclipse.org/org/documents/edl-v10.php.

SPDX-License-Identifier: BSD-3-Clause

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jaxb-ri
* https://github.com/eclipse-ee4j/jaxb-istack-commons
* https://github.com/eclipse-ee4j/jaxb-dtd-parser
* https://github.com/eclipse-ee4j/jaxb-fi
*
 https://github.com/eclipse-ee4j/jaxb-stax-ex
* https://github.com/eclipse-ee4j/jax-rpc-ri

## Third-party Content

This project leverages the following third party content.

Apache Ant (1.10.2)

* License: Apache-2.0 AND W3C AND LicenseRef-Public-Domain

Apache Ant (1.10.2)

* License: Apache-2.0 AND W3C AND LicenseRef-Public-Domain

Apache Felix (1.2.0)

* License: Apache License, 2.0

args4j (2.33)

* License: MIT License

dom4j (1.6.1)

* License: Custom license based on Apache 1.1

file-management (3.0.0)

* License: Apache-2.0
* Project: https://maven.apache.org/shared/file-management/
* Source:
  https://svn.apache.org/viewvc/maven/shared/tags/file-management-3.0.0/

JUnit (4.12)

* License: Eclipse Public License

JUnit (4.12)

* License: Eclipse Public License

maven-compat (3.5.2)

* License: Apache-2.0
* Project: https://maven.apache.org/ref/3.5.2/maven-compat/
* Source:
  https://mvnrepository.com/artifact/org.apache.maven/maven-compat/3.5.2

maven-core
(3.5.2)

* License: Apache-2.0
* Project: https://maven.apache.org/ref/3.5.2/maven-core/index.html
* Source: https://mvnrepository.com/artifact/org.apache.maven/maven-core/3.5.2

maven-plugin-annotations (3.5)

* License: Apache-2.0
* Project: https://maven.apache.org/plugin-tools/maven-plugin-annotations/
* Source:
  https://github.com/apache/maven-plugin-tools/tree/master/maven-plugin-annotations

maven-plugin-api (3.5.2)

* License: Apache-2.0

maven-resolver-api (1.1.1)

* License: Apache-2.0

maven-resolver-api (1.1.1)

* License: Apache-2.0

maven-resolver-connector-basic (1.1.1)

* License: Apache-2.0

maven-resolver-impl (1.1.1)

* License: Apache-2.0

maven-resolver-spi (1.1.1)

* License: Apache-2.0

maven-resolver-transport-file (1.1.1)

* License: Apache-2.0
* Project: https://maven.apache.org/resolver/maven-resolver-transport-file/
* Source:
  https://github.com/apache/maven-resolver/tree/master/maven-resolver-transport-file

maven-resolver-util (1.1.1)

* License: Apache-2.0

maven-settings
(3.5.2)

* License: Apache-2.0
* Source:
  https://mvnrepository.com/artifact/org.apache.maven/maven-settings/3.5.2

OSGi Service Platform Core Companion Code (6.0)

* License: Apache License, 2.0

plexus-archiver (3.5)

* License: Apache-2.0
* Project: https://codehaus-plexus.github.io/plexus-archiver/

* Source: https://github.com/codehaus-plexus/plexus-archiver

plexus-io (3.0.0)

* License: Apache-2.0

plexus-utils (3.1.0)

* License: Apache- 2.0 or Apache- 1.1 or BSD or Public Domain or Indiana
  University Extreme! Lab Software License V1.1.1 (Apache 1.1 style)

relaxng-datatype (1.0)

* License: New BSD license

Sax (0.2)

* License: SAX-PD
* Project: http://www.megginson.com/downloads/SAX/
* Source: http://sourceforge.net/project/showfiles.php?group_id=29449

testng (6.14.2)

* License: Apache-2.0 AND (MIT OR GPL-1.0+)
* Project: https://testng.org/doc/index.html
* Source: https://github.com/cbeust/testng

wagon-http-lightweight (3.0.0)

* License: Pending
*
 Project: https://maven.apache.org/wagon/
* Source:
  https://mvnrepository.com/artifact/org.apache.maven.wagon/wagon-http-lightweight/3.0.0

xz for java (1.8)

* License: LicenseRef-Public-Domain

## Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

# 1.18 antlr-runtime 4.7.1

# 1.19 auto-value-annotations 1.6.2

# 1.20 rxjava 2.2.6

# 1.21 joda-time 2.10.3

## 1.21.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or
(iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant

of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as
of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your
sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==========
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
==========

This product includes software developed by
Joda.org (https://www.joda.org/).

# 1.22 annotations 4.1.1.4

# 1.23 commonmark 0.11.0

# 1.24 jmustache 1.14

# 1.25 rhino 1.7.7.2

## 1.25.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/* -*- Mode: java; tab-width: 8; indent-tabs-mode: nil; c-basic-offset: 4 -*-
*
* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */
/**
* This is interface defines a protocol for the reporting of
* errors during JavaScript translation in IDE-mode.
* If the {@link org.mozilla.javascript.Parser}'s error reporter is
* set to an instance of this interface, then this interface's
* {@link #warning} and {@link #error} methods are called instead
* of the {@link org.mozilla.javascript.ErrorReporter} versions. <p>
*
* These methods take a source char offset and a length.  The
* rationale is that in interactive IDE-type environments, the source
* is available and the IDE will want to indicate where the error
* occurred and how much code participates in it.  The start
 and length
* are generally chosen to fit within a single line, for readability,
* but the client is free to use the AST to determine the affected
* node(s) from the start position and change the error or warning's
* display bounds.<p>
*
* @author Steve Yegge
*/
```

Found in path(s):
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-

jar/org/mozilla/javascript/ast/IdeErrorReporter.java
No license file was found, but licenses were detected in source scan.

```
/*
* @(#)TreeTableModelAdapter.java       1.2 98/10/27
*
* Copyright 1997, 1998 Sun Microsystems, Inc.  All Rights Reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
*   - Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*
*   - Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*
*   - Neither the name of Sun Microsystems nor the names of its
*     contributors may be used to endorse or promote products derived
*     from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS
* IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/debugger/treetable/TreeTableModelAdapter.java
No license file was found, but licenses were detected in source scan.

```
/* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */
```

Found in path(s):
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-

jar/org/mozilla/javascript/optimizer/Optimizer.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/annotations/JSConstructor.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/annotations/JSFunction.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/annotations/JSStaticFunction.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/provider/SoftCachingModuleScriptProvider.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/provider/UrlConnectionExpiryCalculator.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/provider/UrlModuleSourceProvider.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/SecureCaller.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/provider/CachingModuleScriptProviderBase.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/ModuleScript.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/xmlimpl/XmlProcessor.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/optimizer/OptRuntime.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/xmlimpl/XmlNode.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/annotations/JSGetter.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/provider/UrlConnectionSecurityDomainProvider.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/StackStyle.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/RequireBuilder.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/provider/ParsedContentType.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/optimizer/OptFunctionNode.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/optimizer/OptTransformer.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/annotations/JSSetter.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/tools/jsc/Main.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/provider/ModuleSource.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-

jar/org/mozilla/javascript/commonjs/module/Require.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/optimizer/Block.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/provider/ModuleSourceProviderBase.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/ModuleScope.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/ModuleScriptProvider.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/provider/ModuleSourceProvider.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/tools/SourceReader.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/provider/StrongCachingModuleScriptProvider.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/PolicySecurityController.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ScriptStackElement.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/DefiningClassLoader.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/provider/DefaultUrlConnectionExpiryCalculator.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/commonjs/module/provider/MultiModuleScriptProvider.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/tools/shell/ShellConsole.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/RhinoSecurityManager.java
No license file was found, but licenses were detected in source scan.

/* -*- Mode: java; tab-width: 4; indent-tabs-mode: 1; c-basic-offset: 4 -*-
*
* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */

Found in path(s):
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/tools/idswitch/SwitchGenerator.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/IdScriptableObject.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/tools/idswitch/Main.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/NativeMath.java

* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/json/JsonParser.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/idswitch/CodePrinter.java
*
 /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/idswitch/FileBody.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/idswitch/IdValuePair.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeJSON.java
No license file was found, but licenses were detected in source scan.

```
// Copyright 2011 the V8 project authors. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
//     * Redistributions of source code must retain the above copyright
//       notice, this list of conditions and the following disclaimer.
//     * Redistributions in binary form must reproduce the above
//       copyright notice, this list of conditions and the following
//       disclaimer in the documentation and/or other materials provided
//     * Neither the name of Google Inc. nor the names of its
//       from this software without specific prior written permission.
```

Found in path(s):
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/v8dtoa/DoubleConversion.java
No license file was found, but licenses were detected in source scan.

```
/* -*- Mode: java; tab-width: 8; indent-tabs-mode: nil; c-basic-offset: 4 -*-
*
* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */
/*
        * During the great date rewrite of 1.3, we tried to track the
        * evolving ECMA standard, which then had a definition of
        * getYear which always subtracted 1900.  Which we
        * implemented, not realizing that it was incompatible with
        * the old behavior...  now, rather than thrash the behavior
        * yet again, we've decided to leave it with the - 1900
        * behavior and point people to the getFullYear method.  But
        * we try to protect existing scripts that have specified a
        * version...
        */
```

Found in path(s):
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-

jar/org/mozilla/javascript/ContextFactory.java
No license file was found, but licenses were detected in source scan.

/* -*- Mode: java; tab-width: 8; indent-tabs-mode: nil; c-basic-offset: 4 -*-
*
* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */
// throws a special exception. This ensures execution of all pending

Found in path(s):
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeGenerator.java
No license file was found, but licenses were detected in source scan.

// Copyright 2010 the V8 project authors. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
//  * Redistributions of source code must retain the above copyright
//  notice, this list of conditions and the following disclaimer.
//  * Redistributions in binary form must reproduce the above
//  copyright notice, this list of conditions and the following
//  disclaimer in the documentation and/or other materials provided
//  * Neither the name of Google Inc. nor the names of its
//  from this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/v8dtoa/DiyFp.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/v8dtoa/FastDtoa.java
*
 /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/v8dtoa/DoubleHelper.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/v8dtoa/CachedPowers.java
No license file was found, but licenses were detected in source scan.

/* -*- Mode: java; tab-width: 8; indent-tabs-mode: nil; c-basic-offset: 4 -*-
*
* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */
/****************************************************************
 *
 * The author of this software is David M. Gay.
 *
 * Copyright (c) 1991, 2000, 2001 by Lucent Technologies.

```
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose without fee is hereby granted, provided that this entire notice
 * is included in all copies of any software which is or includes a copy
 * or modification of this software and in all copies of the supporting
 * documentation for such software.
 *
 * THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED
 * WARRANTY.  IN PARTICULAR, NEITHER THE AUTHOR
 NOR LUCENT MAKES ANY
 * REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY
 * OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.
 *
 ***************************************************************/
```

Found in path(s):
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/DToA.java
No license file was found, but licenses were detected in source scan.

```
/*
* @(#)JTreeTable.java  1.2 98/10/27
*
* Copyright 1997, 1998 Sun Microsystems, Inc.  All Rights Reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
*   - Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*
*   - Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*
*   - Neither the name of Sun Microsystems nor the names of its
*     contributors may be used to endorse or promote products derived
*     from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
* IS" AND
 ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
```

* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/debugger/treetable/JTreeTable.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 1998 Sun Microsystems, Inc.  All Rights Reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
*   - Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*
*   - Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*
*   - Neither the name of Sun Microsystems nor the names of its
*     contributors may be used to endorse or promote products derived
*     from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
* IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
 BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/debugger/treetable/AbstractCellEditor.java
No license file was found, but licenses were detected in source scan.

/* -*- Mode: java; tab-width: 8; indent-tabs-mode: nil; c-basic-offset: 4 -*-

*
* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */

Found in path(s):
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/XmlElemRef.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/TryStatement.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/debugger/Main.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/shell/ShellLine.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/ConditionalExpression.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ConsString.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/RefCallable.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/jdk13/VMBridge_jdk13.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/classfile/ClassFileWriter.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ScriptRuntimeES6.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Script.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/Scope.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ContextListener.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/CatchClause.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/FunctionCall.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/DoLoop.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeJavaClass.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/ErrorNode.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/EmptyExpression.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/AstRoot.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/BoundFunction.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/regexp/NativeRegExp.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/xmlimpl/XMLObjectImpl.java

*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/RegExpProxy.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ES6Iterator.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/DefaultErrorReporter.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/UnaryExpression.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/regexp/RegExpImpl.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeWith.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/typedarrays/NativeDataView.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/xmlimpl/QName.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/Label.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/ScriptNode.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeString.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/regexp/SubString.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/GeneratorExpressionLoop.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/ErrorCollector.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Synchronizer.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/Symbol.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/SpecialRef.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/RhinoException.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/typedarrays/NativeFloat64Array.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ObjArray.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/TopLevel.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/ArrayComprehension.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/Yield.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ImporterTopLevel.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/SecurityController.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Delegator.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-

jar/org/mozilla/javascript/NativeArrayIterator.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Ref.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/XmlRef.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/tools/shell/SecurityProxy.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/Loop.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ast/VariableInitializer.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ClassShutter.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/NativeJavaConstructor.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/tools/shell/Environment.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/NativeSymbol.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ast/XmlPropRef.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/tools/debugger/ScopeProvider.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/MemberBox.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/v8dtoa/FastDtoaBuilder.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeCall.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/xmlimpl/XMLCtor.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ast/PropertyGet.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/NativeBoolean.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ScriptRuntime.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ast/XmlLiteral.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/NativeCallSite.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/NativeObject.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ast/ForLoop.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/WrappedException.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeDate.java
*

/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeJavaObject.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/NewExpression.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeError.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/typedarrays/NativeInt8Array.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ConstProperties.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/typedarrays/NativeArrayBufferView.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeJavaTopPackage.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/CodeGenerator.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ExternalArrayData.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/xmlimpl/XMLLibImpl.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/xmlimpl/XMLWithScope.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/IfStatement.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/ToolErrorReporter.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/WithStatement.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/optimizer/Codegen.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/shell/QuitAction.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/typedarrays/Conversions.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/typedarrays/NativeUint16Array.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Symbol.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/serialize/ScriptableInputStream.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/UniqueTag.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/ParseProblem.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/InterpretedFunction.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Function.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Arguments.java
*

/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/FunctionNode.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/Assignment.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/WhileLoop.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeJavaMethod.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/EcmaError.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/Block.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/CompilerEnvirons.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeJavaPackage.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Node.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ContextAction.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/typedarrays/NativeFloat32Array.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ClassCache.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/VMBridge.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/LazilyLoadedCtor.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/classfile/ByteCode.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/ThrowStatement.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Icode.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/XmlMemberGet.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/shell/Main.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/UintMap.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Scriptable.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/regexp/NativeRegExpCtor.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/typedarrays/NativeArrayBuffer.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/debug/Debugger.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeIterator.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/JavaAdapter.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ArrowFunction.java

* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeFunction.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/typedarrays/ByteIo.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/Name.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/EvaluatorException.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/LetNode.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/InterfaceAdapter.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/typedarrays/NativeUint8Array.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ContinuationPending.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/SwitchStatement.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/BaseFunction.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/debugger/Dim.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Evaluator.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/KeywordLiteral.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/ParenthesizedExpression.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Decompiler.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/debugger/SourceProvider.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/shell/Global.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Kit.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Callable.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/XmlFragment.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/ForInLoop.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/XmlString.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/debug/DebuggableScript.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/shell/JSConsole.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/debug/DebuggableObject.java
*

/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeStringIterator.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Wrapper.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/DestructuringForm.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/NodeVisitor.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeNumber.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/ArrayLiteral.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/AstNode.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Parser.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/NumberLiteral.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NodeTransformer.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/serialize/ScriptableOutputStream.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeScript.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/shell/ConsoleTextArea.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/xmlimpl/XMLList.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/XmlDotQuery.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/XmlExpression.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/ElementGet.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/JavaScriptException.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ErrorReporter.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/ArrayComprehensionLoop.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/shell/ShellContextFactory.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeGlobal.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/BreakStatement.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/VariableDeclaration.java

* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/typedarrays/NativeUint32Array.java
*
 /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/jdk15/VMBridge_jdk15.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/IdFunctionObject.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/IRFactory.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/ObjectLiteral.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/EmptyStatement.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ScriptableObject.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/InfixExpression.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/typedarrays/NativeTypedArrayView.java
*
 /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/Comment.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/StringLiteral.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/debugger/SwingGui.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/xmlimpl/XML.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/SecurityUtilities.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/optimizer/ClassCompiler.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/NativeJavaArray.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/TokenStream.java
*
 /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/SwitchCase.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/GeneratorExpression.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/xmlimpl/XMLName.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/tools/debugger/GuiCallback.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/typedarrays/NativeInt16Array.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Interpreter.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-

jar/org/mozilla/javascript/ast/ExpressionStatement.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ast/RegExpLiteral.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/NativeContinuation.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/xml/XMLLib.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/GeneratedClassLoader.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/tools/shell/JavaPolicySecurity.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/JavaMembers.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ast/LabeledStatement.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/FunctionObject.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Context.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ast/ReturnStatement.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ObjToIntMap.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/typedarrays/NativeTypedArrayIterator.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ast/ContinueStatement.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/ast/ObjectProperty.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/debug/DebugFrame.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/InterpreterData.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/ast/Jump.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/SymbolScriptable.java
*
/opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/typedarrays/NativeUint8ClampedArray.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/IdFunctionCall.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/WrapFactory.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/typedarrays/NativeInt32Array.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Undefined.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-

jar/org/mozilla/javascript/NativeArray.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/xmlimpl/Namespace.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/xml/XMLObject.java
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-jar/org/mozilla/javascript/Token.java
No license file was found, but licenses were detected in source scan.

```
/*
* TreeTableModel.java
*
* Copyright 1998 Sun Microsystems, Inc.  All Rights Reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
*   - Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*
*   - Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*
*   - Neither the name of Sun Microsystems nor the names of its
*     contributors may be used to endorse or promote products derived
*     from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
* IS" AND ANY EXPRESS OR IMPLIED
 WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/1005120616_1606874013.46/0/rhino-1-7-7-2-sources-
jar/org/mozilla/javascript/tools/debugger/treetable/TreeTableModel.java

# 1.26 converter-scalars 2.5.0

# 1.27 converter-gson 2.5.0

# 1.28 handlebars-jackson2 4.1.2

# 1.29 threetenbp 1.4.0

## 1.29.1 Available under license :

```
/*
* Copyright (c) 2007-present, Stephen Colebourne & Michael Nascimento Santos
*
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*
*  * Redistributions of source code must retain the above copyright notice,
*    this list of conditions and the following disclaimer.
*
*  * Redistributions in binary form must reproduce the above copyright notice,
*    this list of conditions and the following disclaimer in the documentation
*    and/or other materials provided with the distribution.
*
*  * Neither the name of JSR-310 nor the names of its contributors
*    may be used to endorse or promote products derived from this software
*    without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES
 OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
```

* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

# 1.30 listenablefuture 1.0

## 1.30.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1012932365_1591900514.54/0/listenablefuture-1-0-sources-jar/com/google/common/util/concurrent/ListenableFuture.java

# 1.31 jakarta-validation-api 2.0.2

## 1.31.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of

fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form
shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely
link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean
Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute,
all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum
to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use
the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage,

computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor
harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.32 reactive-streams 1.0.3

## 1.32.1 Available under license :

Public Domain

# 1.33 javac-shaded 9-dev-r4023-3

## 1.33.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.)
You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program).

Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the

entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation
of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution and only if you
received the program in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a

special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering

access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to

apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author

to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This
is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than 'show w' and 'show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General
Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

# 1.34 google-java-format 1.5

## 1.34.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2016 Google Inc. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
```

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/RemoveUnusedImports.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/JavaOutput.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/JavaCommentsHelper.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/FormatFileCallable.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/TypeNameClassifier.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/JavaInput.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/Output.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/CommentsHelper.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/Doc.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/Formatter.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/DocBuilder.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/JavaInputAstVisitor.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-

jar/com/google/googlejavaformat/FormatterDiagnostic.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/OpsBuilder.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/Main.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/Op.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/UsageException.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/GoogleJavaFormatVersion.java
*
 /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/FormatterException.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/Indent.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/InputOutput.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/CloseOp.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/Input.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/OpenOp.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/SnippetFormatter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 Google Inc.

*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/Trees.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/Newlines.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/CommandLineOptionsParser.java
*
 /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/javadoc/Token.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/javadoc/JavadocFormatter.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/FormattingError.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/javadoc/JavadocWriter.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/JavacTokens.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/DimensionHelpers.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/ImportOrderer.java
*
 /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/javadoc/JavadocLexer.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/filer/FormattingFiler.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/javadoc/CharStream.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/filer/FormattingJavaFileObject.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/javadoc/NestingCounter.java
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-
jar/com/google/googlejavaformat/java/JavaFormatterOptions.java

*
/opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/CommandLineOptions.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2016 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not
* use this file except in compliance with the License. You may obtain a copy
* of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/ModifierOrderer.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 Google Inc. All rights reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/Replacement.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2015 Google Inc.
```

```
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

package com.google.googlejavaformat.java;

import java.util.Optional;

class GoogleJavaFormatVersion {

 static String version() {
   return "%VERSION%";
 }
}
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1015771199_1594201050.18/0/google-java-format-1-5-sources-jar/com/google/googlejavaformat/java/GoogleJavaFormatVersion.java.template

# 1.35 slf4j-simple 1.7.29

## 1.35.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/**
* Copyright (c) 2004-2011 QOS.ch
* All rights reserved.
*
* Permission is hereby granted, free  of charge, to any person obtaining
* a  copy  of this  software  and  associated  documentation files  (the
* "Software"), to  deal in  the Software without  restriction, including
* without limitation  the rights to  use, copy, modify,  merge, publish,
* distribute,  sublicense, and/or sell  copies of  the Software,  and to
* permit persons to whom the Software  is furnished to do so, subject to
* the following conditions:
*
* The  above  copyright  notice  and  this permission  notice  shall  be
* included in all copies or substantial portions of the Software.
```

*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1016414134_1591383926.05/0/slf4j-simple-1-7-29-sources-jar/org/slf4j/impl/SimpleLoggerFactory.java
* /opt/ws_local/PERMITS_SQL/1016414134_1591383926.05/0/slf4j-simple-1-7-29-sources-jar/org/slf4j/impl/StaticMDCBinder.java
* /opt/ws_local/PERMITS_SQL/1016414134_1591383926.05/0/slf4j-simple-1-7-29-sources-jar/org/slf4j/impl/StaticLoggerBinder.java
* /opt/ws_local/PERMITS_SQL/1016414134_1591383926.05/0/slf4j-simple-1-7-29-sources-jar/org/slf4j/impl/StaticMarkerBinder.java
No license file was found, but licenses were detected in source scan.

/**
* Copyright (c) 2004-2012 QOS.ch
* All rights reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT
SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1016414134_1591383926.05/0/slf4j-simple-1-7-29-sources-jar/org/slf4j/impl/SimpleLogger.java

# 1.36 slf4j-ext 1.7.29

## 1.36.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.37 slf4j 1.7.30

## 1.37.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.38 commons-compress 1.20

## 1.38.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Commons Compress
Copyright 2002-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

---

The files in the package org.apache.commons.compress.archivers.sevenz
were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/),
which has been placed in the public domain:

"LZMA SDK is placed in the public domain." (http://www.7-zip.org/sdk.html)

---

The test file lbzip2_32767.bz2 has been copied from libbzip2's source repository:

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2019 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, jseward@acm.org

# 1.39 fast-infoset 1.2.16

## 1.39.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# Notices for Eclipse Implementation of JAXB

This content is produced and maintained by the Eclipse Implementation of JAXB
project.

* Project home: https://projects.eclipse.org/projects/ee4j.jaxb-impl

## Trademarks

Eclipse Implementation of JAXB is a trademark of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For
more information regarding authorship of content, please consult the listed
source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms
of the Eclipse Distribution License v. 1.0 which is available at
http://www.eclipse.org/org/documents/edl-v10.php.

SPDX-License-Identifier: BSD-3-Clause

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jaxb-ri
* https://github.com/eclipse-ee4j/jaxb-istack-commons
* https://github.com/eclipse-ee4j/jaxb-dtd-parser
* https://github.com/eclipse-ee4j/jaxb-fi
*
 https://github.com/eclipse-ee4j/jaxb-stax-ex
* https://github.com/eclipse-ee4j/jax-rpc-ri

## Third-party Content

This project leverages the following third party content.

Apache Ant (1.10.2)

* License: Apache-2.0 AND W3C AND LicenseRef-Public-Domain

Apache Ant (1.10.2)

* License: Apache-2.0 AND W3C AND LicenseRef-Public-Domain

Apache Felix (1.2.0)

* License: Apache License, 2.0

args4j (2.33)

* License: MIT License

dom4j (1.6.1)

* License: Custom license based on Apache 1.1

file-management (3.0.0)

* License: Apache-2.0
* Project: https://maven.apache.org/shared/file-management/
* Source:
  https://svn.apache.org/viewvc/maven/shared/tags/file-management-3.0.0/

JUnit (4.12)

* License: Eclipse Public License

JUnit (4.12)

* License: Eclipse Public License

maven-compat (3.5.2)

* License: Apache-2.0
* Project: https://maven.apache.org/ref/3.5.2/maven-compat/
* Source:
  https://mvnrepository.com/artifact/org.apache.maven/maven-compat/3.5.2

maven-core
 (3.5.2)

* License: Apache-2.0
* Project: https://maven.apache.org/ref/3.5.2/maven-core/index.html
* Source: https://mvnrepository.com/artifact/org.apache.maven/maven-core/3.5.2

maven-plugin-annotations (3.5)

* License: Apache-2.0
* Project: https://maven.apache.org/plugin-tools/maven-plugin-annotations/
* Source:
  https://github.com/apache/maven-plugin-tools/tree/master/maven-plugin-annotations

maven-plugin-api (3.5.2)

* License: Apache-2.0

maven-resolver-api (1.1.1)

* License: Apache-2.0

maven-resolver-api (1.1.1)

* License: Apache-2.0

maven-resolver-connector-basic (1.1.1)

* License: Apache-2.0

maven-resolver-impl (1.1.1)

* License: Apache-2.0

maven-resolver-spi (1.1.1)

* License: Apache-2.0

maven-resolver-transport-file (1.1.1)

* License: Apache-2.0
* Project: https://maven.apache.org/resolver/maven-resolver-transport-file/

* Source:
  https://github.com/apache/maven-resolver/tree/master/maven-resolver-transport-file

maven-resolver-util (1.1.1)

* License: Apache-2.0

maven-settings
 (3.5.2)

* License: Apache-2.0
* Source:
  https://mvnrepository.com/artifact/org.apache.maven/maven-settings/3.5.2

OSGi Service Platform Core Companion Code (6.0)

* License: Apache License, 2.0

plexus-archiver (3.5)

* License: Apache-2.0
* Project: https://codehaus-plexus.github.io/plexus-archiver/
* Source: https://github.com/codehaus-plexus/plexus-archiver

plexus-io (3.0.0)

* License: Apache-2.0

plexus-utils (3.1.0)

* License: Apache- 2.0 or Apache- 1.1 or BSD or Public Domain or Indiana
  University Extreme! Lab Software License V1.1.1 (Apache 1.1 style)

relaxng-datatype (1.0)

* License: New BSD license

Sax (0.2)

* License: SAX-PD
* Project: http://www.megginson.com/downloads/SAX/
* Source: http://sourceforge.net/project/showfiles.php?group_id=29449

testng (6.14.2)

* License: Apache-2.0 AND (MIT OR GPL-1.0+)
* Project: https://testng.org/doc/index.html
* Source: https://github.com/cbeust/testng

wagon-http-lightweight (3.0.0)

* License: Pending
*
 Project: https://maven.apache.org/wagon/
* Source:
  https://mvnrepository.com/artifact/org.apache.maven.wagon/wagon-http-lightweight/3.0.0

xz for java (1.8)

* License: LicenseRef-Public-Domain

## Cryptography

Content may contain encryption software. The country in which you are currently
may have restrictions on the import, possession, and use, and/or re-export to
another country, of encryption software. BEFORE using any encryption software,
please check the country's laws, regulations and policies concerning the import,
possession, or use, and re-export of encryption software, to see if this is
permitted.

Apache ServiceMix :: Bundles :: FastInfoset
Copyright 2005-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

# 1.40 checker-qual 2.11.1

## 1.40.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to

using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.)
You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a

notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this
License. (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)

These requirements apply to the modified
work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation
of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution and only if you

received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a

special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author

to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option)
any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This
is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than 'show w' and 'show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General
Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

# 1.41 caffeine 2.8.1

## 1.41.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2017 Ben Manes. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/Expiry.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/TimerWheel.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2019 Ben Manes. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/Pacer.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/Scheduler.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2018 Ben Manes. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/AsyncCache.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/LocalAsyncCache.java
No license file was found, but licenses were detected in source scan.

```
/*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/MpscGrowableArrayQueue.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2015 Ben Manes. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
```

* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/stats/GuardedStatsCounter.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/CacheWriter.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WriteThroughEntry.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/Node.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/LocalAsyncLoadingCache.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/Buffer.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FrequencySketch.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SerializationProxy.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/stats/package-info.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/Async.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/package-info.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/LocalManualCache.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/package-info.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/BoundedBuffer.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/base/package-info.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/LocalLoadingCache.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-

jar/com/github/benmanes/caffeine/cache/LocalCache.java
*
 /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-
jar/com/github/benmanes/caffeine/cache/References.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2015 Ben Manes. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-
jar/com/github/benmanes/caffeine/cache/StripedBuffer.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2016 Ben Manes. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/CaffeineSpec.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/AsyncCacheLoader.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 Ben Manes. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/stats/DisabledStatsCounter.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/Ticker.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/LinkedDeque.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/UnboundedLocalCache.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/Weigher.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/Cache.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/CacheLoader.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WriteOrderDeque.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/SingleConsumerQueue.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/Caffeine.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/RemovalListener.java

* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/RemovalCause.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/base/UnsafeAccess.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/BoundedLocalCache.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/AccessOrderDeque.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/LoadingCache.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/stats/StatsCounter.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/AsyncLoadingCache.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/AbstractLinkedDeque.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/stats/ConcurrentStatsCounter.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/stats/CacheStats.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/Policy.java
No license file was found, but licenses were detected in source scan.

// Copyright 2020 Ben Manes. All Rights Reserved.
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//  http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWAWMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMWR.java

* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWAWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSAWRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMWA.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDAMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSAWMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMWAWR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-

jar/com/github/benmanes/caffeine/cache/SISMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDAWMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMSW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDRMS.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMS.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMSR.java

* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIAR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIL.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMWAWR.java

*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSWMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMSW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDWMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWAWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMS.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-

jar/com/github/benmanes/caffeine/cache/PSWR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWWRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMSAWR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDWRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSWRMW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWAMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILS.java

* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSAW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDAWMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMSAR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-

jar/com/github/benmanes/caffeine/cache/SSLMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDAWMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDARMS.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWAWMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDWMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSARMS.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWAWMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDARMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMSAWR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLA.java

* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWARMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWARMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-

jar/com/github/benmanes/caffeine/cache/WSLAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSAWMS.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDAR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWWRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/LocalCacheFactory.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/NodeFactory.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIAW.java

* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWAMW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDAWRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-

jar/com/github/benmanes/caffeine/cache/WILAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDARMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMWAR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FD.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMSR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWWMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMWAR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWARMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMSW.java

* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSWMW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSARMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSRMS.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMSW.java
*

/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMSWR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSL.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWWMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSAMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDWMS.java

*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSAR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSL.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-

jar/com/github/benmanes/caffeine/cache/FWRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDWMS.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWARMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMWW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDAWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMWAW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDWRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLWR.java

* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWA.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWAW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSAWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMSAWR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-

jar/com/github/benmanes/caffeine/cache/WSLSMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMSAWR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSARMW.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWAMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDAMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMSA.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLR.java

* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSAWMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDAWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWAWRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWWMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSAWMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-

jar/com/github/benmanes/caffeine/cache/FDA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSAWRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDAWMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSAMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWAWMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSAW.java

* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWWMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDAMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIL.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSAWRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-

jar/com/github/benmanes/caffeine/cache/WISAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILSMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSSMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDARMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWAWRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWAMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMSWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSAMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SI.java

* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSWMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSWMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISA.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSARMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSAMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMSWR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILMSAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMWW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDAMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WI.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSWRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMWR.java

*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLSMSAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PD.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMSR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FSRMS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIMWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FDAWRMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/FWWR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PSA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMSAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PDMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/PW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WILMWA.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SIMSAR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSSMSW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSMWAW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSMWWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SSLMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SISMWAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISMW.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-

jar/com/github/benmanes/caffeine/cache/WISMWAWR.java
*
/opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WIS.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WISAR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/WSLSMWAWR.java
* /opt/ws_local/PERMITS_SQL/1045925391_1589677258.18/0/caffeine-2-8-1-sources-jar/com/github/benmanes/caffeine/cache/SILSMW.java

# 1.42 httpcore 4.4.13

## 1.42.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* ========================================================================
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements.  See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership.  The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License.  You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied.  See the License for the
* specific language governing permissions and limitations
* under the License.
* ========================================================================
 *
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation.  For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/entity/DisallowIdentityContentLengthStrategy.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-

jar/org/apache/httpcore/message/BasicHeaderValueFormatter.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/bootstrap/ThreadFactoryImpl.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpEntity.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/util/VersionInfo.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/MessageConstraintException.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpConnection.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicHeader.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/pool/PoolEntryCallback.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/annotation/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/ChunkedInputStream.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/AbstractMessageWriter.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/pool/PoolStats.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ContentTooLongException.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/DefaultBHttpServerConnection.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/ContentLengthOutputStream.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/DefaultHttpResponseWriter.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/NoHttpResponseException.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HeaderElementIterator.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/DefaultHttpRequestWriter.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ExceptionLogger.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/AbstractHttpEntity.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HttpRequestHandlerMapper.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/HeaderGroup.java

* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/Header.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/StatusLine.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ReasonPhraseCatalog.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ConnectionClosedException.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/config/SocketConfig.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/params/CoreProtocolPNames.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HttpService.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/annotation/Experimental.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/UnsupportedHttpVersionException.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/ResponseDate.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ssl/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicHttpResponse.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/EmptyInputStream.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/BHttpConnectionBase.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/config/Registry.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/StringEntity.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/entity/StrictContentLengthStrategy.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/params/BasicHttpParams.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HttpResponseInterceptorList.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/SessionInputBufferImpl.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/config/MessageConstraints.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/pool/BasicPoolEntry.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-

jar/org/apache/httpcore/params/AbstractHttpParams.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/TruncatedChunkException.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HttpExpectationVerifier.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HttpCoreContext.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/FormattedHeader.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/HttpEntityWrapper.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicLineFormatter.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicHeaderElement.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/io/HttpTransportMetrics.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/bootstrap/Worker.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/TokenIterator.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpEntityEnclosingRequest.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/DefaultHttpRequestParserFactory.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/DefaultHttpResponseParser.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HttpRequestExecutor.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/FileEntity.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ssl/TrustStrategy.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/RequestUserAgent.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/InputStreamEntity.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/params/HttpParams.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/annotation/Contract.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/BufferedHttpEntity.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/ByteArrayEntity.java

* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicHttpEntityEnclosingRequest.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ParseException.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/NoConnectionReuseStrategy.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/util/NetUtils.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/bootstrap/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/EntityTemplate.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/io/HttpMessageParserFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/DefaultHttpResponseWriterFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/config/ConnectionConfig.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/io/SessionOutputBuffer.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpVersion.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/config/Lookup.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/ContentType.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/pool/ConnPoolControl.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicHeaderValueParser.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ConnectionReuseStrategy.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/BasicHttpEntity.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/bootstrap/RequestListener.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/entity/LaxContentLengthStrategy.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/util/TextUtils.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpResponse.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/DefaultHttpResponseFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-

jar/org/apache/httpcore/impl/entity/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/DefaultBHttpServerConnectionFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpInetConnection.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/annotation/ThreadingBehavior.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/SerializableEntity.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/ChunkedOutputStream.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/params/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/NameValuePair.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HttpRequestHandlerResolver.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicLineParser.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/util/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/UriHttpRequestHandlerMapper.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HttpRequestHandler.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/params/CoreConnectionPNames.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/ImmutableHttpProcessor.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/concurrent/BasicFuture.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ssl/PrivateKeyStrategy.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/RequestTargetHost.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/annotation/Obsolete.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/ContentLengthStrategy.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpHeaders.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpRequestInterceptor.java

* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/ResponseConnControl.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicTokenIterator.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicStatusLine.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/pool/AbstractConnPool.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/util/EncodingUtils.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/DefaultHttpRequestFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/Consts.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/IdentityInputStream.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/pool/ConnFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpConnectionMetrics.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/HeaderValueFormatter.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpResponseInterceptor.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ProtocolVersion.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/DefaultHttpRequestWriterFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/ResponseContent.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpResponseFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/DefaultHttpRequestParser.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/pool/RouteSpecificPool.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpRequestFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HttpContext.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/params/HttpParamsNames.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicHeaderElementIterator.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-

jar/org/apache/httpcore/impl/pool/BasicConnFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/concurrent/FutureCallback.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpRequest.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpClientConnection.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/util/CharArrayBuffer.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/concurrent/package-info.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpHost.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/DefaultBHttpClientConnectionFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/params/HttpParamConfig.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/concurrent/Cancellable.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/HttpConnectionMetricsImpl.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/RequestLine.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicRequestLine.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/util/ByteArrayBuffer.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/DefaultHttpResponseParserFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/EnglishReasonPhraseCatalog.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/ParserCursor.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpStatus.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/HeaderValueParser.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/io/HttpMessageParser.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/LineFormatter.java

* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpConnectionFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/util/Asserts.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/io/HttpMessageWriterFactory.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicListHeaderIterator.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/MalformedChunkCodingException.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/util/EntityUtils.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/ResponseServer.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/RequestExpectContinue.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/util/CharsetUtils.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HTTP.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HeaderElement.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HttpProcessor.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/config/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ssl/SSLContexts.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/entity/ContentProducer.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/HttpTransportMetricsImpl.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/pool/ConnPool.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/pool/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/config/RegistryBuilder.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/pool/PoolEntry.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-

jar/org/apache/httpcore/impl/bootstrap/SSLServerSetupHandler.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/IdentityOutputStream.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HttpRequestInterceptorList.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/AbstractMessageParser.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ssl/SSLInitializationException.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/io/BufferInfo.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/SessionOutputBufferImpl.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicNameValuePair.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/TokenParser.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ssl/PrivateKeyDetails.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/ChainBuilder.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/bootstrap/WorkerPoolExecutor.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/util/Args.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/bootstrap/HttpServer.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/DefaultBHttpClientConnection.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/ConnSupport.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/io/ContentLengthInputStream.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/io/SessionInputBuffer.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/MethodNotSupportedException.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HeaderIterator.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/LineParser.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicHttpRequest.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ssl/SSLContextBuilder.java

* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/pool/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/io/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/util/LangUtils.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BufferedHeader.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/RequestConnControl.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/io/HttpMessageWriter.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpException.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/bootstrap/ServerBootstrap.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/AbstractHttpMessage.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HttpProcessorBuilder.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/io/EofSensor.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/HttpDateGenerator.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/package-info.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/UriPatternMatcher.java
*
/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/ProtocolException.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpServerConnection.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/RequestDate.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/BasicHttpContext.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/pool/BasicConnPool.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/message/BasicHeaderIterator.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/HttpMessage.java
* /opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/impl/DefaultConnectionReuseStrategy.java
*

/opt/ws_local/PERMITS_SQL/1052411894_1591996206.29/0/httpcore-4-4-13-sources-jar/org/apache/httpcore/protocol/RequestContent.java

# 1.43 javassist 3.26.0-GA

## 1.43.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Javassist, a Java-bytecode translator toolkit.
* Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.
*
* The contents of this file are subject to the Mozilla Public License Version
* 1.1 (the "License"); you may not use this file except in compliance with
* the License.  Alternatively, the contents of this file may be used under
* the terms of the GNU Lesser General Public License Version 2.1 or later,
* or the Apache License Version 2.0.
*
* Software distributed under the License is distributed on an "AS IS" basis,
* WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
* for the specific language governing rights and limitations under the
* License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/analysis/SubroutineScanner.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/HotSwapper.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/Visitor.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/Mnemonic.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/analysis/Executor.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/CallExpr.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/reflect/Compiler.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/SyntheticAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/AnnotationDefaultAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/stackmap/Tracer.java
*

/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/expr/Instanceof.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/ByteStream.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/URLClassPath.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/Opcode.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/Translator.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/NoSuchClassError.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/proxy/MethodFilter.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/proxy/ProxyObjectOutputStream.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/rmi/Sample.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/Stmnt.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/Pair.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/convert/TransformCall.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/InnerClassesAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/CodeAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/rmi/Proxy.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/analysis/MultiType.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/convert/TransformAfter.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/analysis/ControlFlow.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/analysis/Type.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/LongVector.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/Expr.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/web/Webserver.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/LocalVariableTypeAttribute.java

* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/reflect/ClassMetaobject.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/analysis/Subroutine.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/DoubleConst.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/ExceptionsAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/expr/NewArray.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/reflect/CannotCreateException.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/Descriptor.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/SymbolTable.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/Javac.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/NestMembersAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/analysis/MultiArrayType.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/analysis/IntQueue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/MemberResolver.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/stackmap/MapMaker.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtPrimitiveType.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/ConstPool.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/convert/TransformFieldAccess.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/AnnotationImpl.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/runtime/Cflow.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/SourceFileAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/CodeAnalyzer.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/AssignExpr.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-

jar/javassist/compiler/ast/Variable.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/Symbol.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/BadBytecode.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/MethodInfo.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/EnclosingMethodAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/ClassFileWriter.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/ByteArray.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/runtime/DotClass.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/rmi/StubGenerator.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/convert/TransformNew.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/proxy/ProxyObject.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/convert/TransformBefore.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/AccessFlag.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/scopedpool/ScopedClassPoolFactoryImpl.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/ClassClassPath.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/analysis/Util.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/CodeIterator.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/Dump.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/ClassPool.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CannotCompileException.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/ASTree.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/analysis/Analyzer.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/convert/TransformNewClass.java

* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/reflect/Metaobject.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/StackMap.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/ClassPath.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/reflect/CannotInvokeException.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/expr/Handler.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/AttributeInfo.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtClass.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/scopedpool/ScopedClassPoolRepository.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CodeConverter.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/stackmap/TypeData.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/ParameterAnnotationsAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/Callback.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/reflect/Loader.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtNewClass.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/expr/ConstructorCall.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/web/Viewer.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/proxy/SerializedProxy.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtNewConstructor.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtField.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/convert/TransformAccessArrayField.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/ByteArrayClassPath.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/proxy/RuntimeSupport.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-

jar/javassist/expr/NewExpr.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/DeprecatedAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/expr/FieldAccess.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/expr/Expr.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/LoaderClassPath.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/InstructionPrinter.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/AccessorMaker.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/LocalVariableAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/NoFieldException.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/Member.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/proxy/SecurityActions.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/analysis/FramePrinter.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/ClassFilePrinter.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/proxy/Proxy.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/HotSwapAgent.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/NestHostAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/LineNumberAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/AnnotationsAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/ClassPoolTail.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/ClassFile.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/Keyword.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/stackmap/BasicBlock.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/proxy/ProxyFactory.java

* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/Modifier.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/reflect/Reflection.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/rmi/ObjectNotFoundException.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/stackmap/TypeTag.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/BinExpr.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/SerialVersionUID.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/Loader.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/JvstTypeChecker.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/CodeGen.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/Declarator.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/Parser.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/FieldInfo.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/scopedpool/ScopedClassPoolFactory.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtBehavior.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtNewWrappedConstructor.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/IntConst.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/reflect/Metalevel.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/Bytecode.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtArray.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/TypeChecker.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/StackMapTable.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/runtime/Desc.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/convert/TransformReadField.java

* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/StringL.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/MethodDecl.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/scopedpool/ScopedClassPoolRepositoryImpl.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/rmi/ObjectImporter.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/Lex.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/ExceptionTable.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/InstanceOfExpr.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/CompileError.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/CastExpr.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/ArrayInit.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/expr/MethodCall.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/NotFoundException.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/framedump.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/proxy/MethodHandler.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/convert/TransformWriteField.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/SignatureAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/rmi/AppletServer.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/AnnotationsWriter.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/web/BadHttpRequest.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/JvstCodeGen.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtMethod.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtMember.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-

jar/javassist/convert/Transformer.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/stackmap/TypedBlock.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/rmi/RemoteException.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/rmi/RemoteRef.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/proxy/ProxyObjectInputStream.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/TokenId.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/analysis/Frame.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/KeywordTable.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/proxy/DefinePackageHelper.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtClassType.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/CondExpr.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/proxy/FactoryHelper.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/scopedpool/ScopedClassPool.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/runtime/Inner.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/DuplicateMemberException.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/util/proxy/DefineClassHelper.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/reflect/Sample.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/MemberCodeGen.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/ASTList.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/ConstantAttribute.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/tools/reflect/CannotReflectException.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtNewMethod.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/expr/Cast.java

* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/expr/ExprEditor.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtConstructor.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/SyntaxError.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/CtNewWrappedMethod.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/NewExpr.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/ClassMap.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/scopedpool/SoftValueHashMap.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ast/FieldDecl.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/compiler/ProceedHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Javassist, a Java-bytecode translator toolkit.
* Copyright (C) 2004 Bill Burke. All Rights Reserved.
*
* The contents of this file are subject to the Mozilla Public License Version
* 1.1 (the "License"); you may not use this file except in compliance with
* the License.  Alternatively, the contents of this file may be used under
* the terms of the GNU Lesser General Public License Version 2.1 or later,
* or the Apache License Version 2.0.
*
* Software distributed under the License is distributed on an "AS IS" basis,
* WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
* for the specific language governing rights and limitations under the
* License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/EnumMemberValue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/Annotation.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/CharMemberValue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/FloatMemberValue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-

jar/javassist/bytecode/annotation/ByteMemberValue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/AnnotationMemberValue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/ShortMemberValue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/DoubleMemberValue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/MemberValue.java
*
/opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/ClassMemberValue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/BooleanMemberValue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/LongMemberValue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/IntegerMemberValue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/ArrayMemberValue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/StringMemberValue.java
* /opt/ws_local/PERMITS_SQL/1068276696_1594339637.94/0/javassist-3-26-0-ga-sources-jar/javassist/bytecode/annotation/MemberValueVisitor.java

# 1.44 jdom 2.0.6

## 1.44.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2011 - 2012 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
derived from this software without prior written permission. For
written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
may "JDOM" appear in their name, without
prior written permission
from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/internal/ReflectionConstructor.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/util/NamespaceStack.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/internal/ArrayCopy.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2011 - 2012 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>.  For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/SAXTarget.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2012 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without prior
 written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED

WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/filter/PassThroughFilter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/filter/TextOnlyFilter.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2011-2012 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission. For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
may "JDOM" appear in their name, without
prior written permission
from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

```
*/
/*
 * ==========================================================================
 * Support methods for Text-content formatting. Should all be protected. The
 * following are used when printing Text-based data. Because of complicated
 * multi-sequential text sometimes the requirements are odd. All Text
 * content will be output using these methods, which is why there is the None
 * version.
 * ==========================================================================

 */
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-

zip/org/jdom2/output/support/AbstractOutputProcessor.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2006 Brett McLaughlin & Jason Hunter.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
 notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
 notice, this list of conditions, and the disclaimer that follows
 these conditions in the documentation and/or other materials
 provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
 derived from this software without prior written permission.  For
 written permission, please contact license@jdom.org.

4. Products derived from this software may not be called "JDOM", nor
 may "JDOM" appear in their name, without prior written permission
 from
 the JDOM Project Management (pm@jdom.org).

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
  "This product includes software developed by the
   JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY
 OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT

OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Brett McLaughlin <brett@jdom.org> and
Jason Hunter <jhunter@jdom.org>. For more information on the
JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/test/cases/TestCDATA.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2000-2011 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission. For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos

available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/DOMOutputter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/BuilderErrorHandler.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2012 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in mtsource and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of mtsource code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
derived from this software without prior written permission. For
written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
may "JDOM" appear in their name, without
prior written permission
from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/AbstractFormattedWalker.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2000 Brett McLaughlin & Jason Hunter.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission. For
   written permission, please contact license@jdom.org.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without prior written
 permission
   from the JDOM Project Management (pm@jdom.org).

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Brett McLaughlin <brett@jdom.org> and
Jason Hunter <jhunter@jdom.org>. For more information on the

JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/test/cases/input/TestSAXBuilder.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/beans/StringConverter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/beans/TestIndexed.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/beans/JDOMBean.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/beans/TestBean.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/output/JTreeOutputter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/helpers/XPathHelper.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/input/scanner/ElementScanner.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/beans/DateUtils.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/input/ResultSetBuilder.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/beans/BeanMapperException.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2000-2007 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions

are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>.  For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

```
/*
 * ==========================================================================
 * Support methods for Text-content formatting. Should all be protected. The
 * following are used when printing Text-based data. Because of complicated
 * multi-sequential text sometimes the requirements are odd. All Text
 * content will be output using these methods, which is why there is the None
 * version.
 * ==========================================================================

 */
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/AbstractXMLOutputProcessor.java
No license file was found, but licenses were detected in source scan.

```
/*--

Copyright (C) 2000-2012 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
```

available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/TextBuffer.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/FilterIterator.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/filter/NegateFilter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/Parent.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/EntityRef.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/ProcessingInstruction.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/AttributeList.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/SAXOutputter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/JDOMFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/Attribute.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/XPathHelper.java

* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/Document.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/filter/ElementFilter.java
*
 /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/jaxen/JDOMXPath.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/SAXBuilder.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/SAXHandler.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/filter/OrFilter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/Format.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/helpers/JDOMHelper.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/filter/ContentFilter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/DescendantIterator.java
*
 /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/CDATA.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/DOMBuilder.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/XMLOutputter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/Comment.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/JDOMParseException.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/DefaultJDOMFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/filter/AndFilter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/Text.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/Element.java
*
 /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/DocType.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/filter/AbstractFilter.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2011-2014 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many

individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>.  For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/perf/SavingVerifier.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2000 Brett McLaughlin & Jason Hunter.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
 notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
 notice, this list of conditions, and the disclaimer that follows
 these conditions in the documentation and/or other materials
 provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
 derived from this software without prior written permission.  For
 written permission, please contact license@jdom.org.

4. Products derived from this software may not be called "JDOM", nor
 may "JDOM" appear in their name, without prior written permission
 from
 the JDOM Project Management (pm@jdom.org).

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
  "This product includes software developed by the
   JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY
OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Brett McLaughlin <brett@jdom.org> and
Jason Hunter <jhunter@jdom.org>. For more information on the
JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/test/cases/TestDocType.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/test/cases/TestVerifier.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/test/cases/TestFilterList.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin & Alex Chaffee.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission. For

written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
may "JDOM" appear in their
name, without prior written permission
from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/beans/BeanMapper.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2011-2014 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>.  For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/perf/DevNull.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/xpath/xalan/XalanXPathExpression.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dtdaware/AttFilter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/perf/TimeRunnable.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/android/TranslateTests.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/perf/PerfVerifier.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/xpath/java/JavaXPathFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/xpath/java/JavaXPathExpression.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/xpath/xalan/XalanXPathFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/perf/PerfTest.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/perf/PerfDoc.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2011-2012 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
may "JDOM" appear in their name, without
prior written permission
from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/filter/Filters.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/NamespaceAware.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/filter/ClassFilter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/JDOMConstants.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2000-2012 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org).

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many

individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>.  For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/ContentList.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2007 Jason Hunter.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
 notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
 notice, this list of conditions, and the disclaimer that follows
 these conditions in the documentation and/or other materials
 provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
 derived from this software without prior written permission.  For
 written permission, please contact license@jdom.org.

4. Products derived from this software may not be called "JDOM", nor
 may "JDOM" appear in their name, without prior written permission
 from the JDOM Project
 Management (pm@jdom.org).

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
  "This product includes software developed by the
   JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER
IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Brett McLaughlin <brett@jdom.org> and
Jason Hunter <jhunter@jdom.org>. For more information on the
JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/test/cases/TestSerialization.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2007-2012 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission. For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without

prior written permission
from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/Content.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2012 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright

notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions, and the disclaimer that follows
these conditions in the documentation and/or other materials
provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
derived from this software without prior written permission. For
written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
may "JDOM" appear in their name, without prior
written permission
from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-

zip/org/jdom2/located/LocatedJDOMFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/located/LocatedProcessingInstruction.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/located/LocatedText.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/WalkerTRIM_FULL_WHITE.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/Walker.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/located/Located.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/located/LocatedDocType.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/WalkerTRIM.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/located/LocatedComment.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dtdaware/AttFilteredXMLOutputProcessor.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/WalkerPRESERVE.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/WalkerNORMALIZE.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/located/LocatedElement.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dtdaware/AttAwareXMLOutputProcessor.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/located/LocatedCDATA.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/located/LocatedEntityRef.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2003-2007 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright

notice, this list of conditions, and the disclaimer that follows
these conditions in the documentation and/or other materials
provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
derived from this software without prior written permission. For
written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
may "JDOM" appear in their name, without
prior written permission
from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/transform/XSLTransformException.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2011-2012 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>.  For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/package-info.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/AttributeType.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/StAXEventBuilder.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/jaxen/JDOMNavigator.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/stax/DTDParser.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/StAXStreamBuilder.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/StAXEventOutputter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/jaxen/NamespaceContainer.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/StAXStreamOutputter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/filter/AttributeFilter.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2003-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
derived from this software without prior written permission. For
written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
may "JDOM" appear in their name, without
prior written permission
from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/schema/ValidationError.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/schema/Schema.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2011 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without prior
 written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many

individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>.  For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/AbstractReaderXSDFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/SAXEngine.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/stax/DefaultStAXFilter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/AbstractReaderSchemaFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/XMLReaderJDOMFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/StringBin.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/stax/StAXFilter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/XMLReaderXSDFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/XMLReaderSchemaFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/XMLReaders.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/XMLReaderSAX2Factory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/SAXHandlerFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/SAXBuilderEngine.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2000-2012 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright

notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions, and the disclaimer that follows
these conditions in the documentation and/or other materials
provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
derived from this software without prior written permission. For
written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
may "JDOM" appear in their name, without
prior written permission
from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-

zip/org/jdom2/DataConversionException.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/Verifier.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/filter/Filter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/JDOMException.java
*
/opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/Namespace.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/adapters/AbstractDOMAdapter.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/IllegalAddException.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/StAXStreamProcessor.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/IllegalDataException.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/JDOMLocator.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/IllegalNameException.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/adapters/JAXPDOMAdapter.java
*
/opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/IllegalTargetException.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/UncheckedJDOMFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/verifier/VerifierBuilder.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/adapters/DOMAdapter.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2001-2007 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright

notice, this list of conditions, and the disclaimer that follows
these conditions in the documentation and/or other materials
provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
derived from this software without prior written permission. For
written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
may "JDOM" appear in their name, without
prior written permission
from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/transform/XSLTransformer.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2001-2007 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>.  For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/transform/JDOMResult.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/transform/JDOMSource.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2000-2007 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/DOMOutputProcessor.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/XMLOutputProcessor.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/SAXOutputProcessor.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/AbstractDOMOutputProcessor.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/StAXEventProcessor.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/FormatStack.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/AbstractStAXEventProcessor.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/AbstractStAXStreamProcessor.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2000-2007 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>.  For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/support/AbstractSAXOutputProcessor.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/XPath.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without prior
 written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/input/LineNumberSAXHandler.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/input/LineNumberElement.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2000 Brett McLaughlin & Jason Hunter.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
 notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
 notice, this list of conditions, and the disclaimer that follows
 these conditions in the documentation and/or other materials
 provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
 derived from this software without prior written permission. For
 written permission, please contact license@jdom.org.

4. Products derived from this software may not be called "JDOM", nor
 may "JDOM" appear in their name, without prior written permission

from the
JDOM Project Management (pm@jdom.org).

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Brett McLaughlin <brett@jdom.org> and
Jason Hunter <jhunter@jdom.org>. For more information on the
JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/test/cases/TestComment.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/test/cases/TestNamespace.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2001-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>.  For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/input/scanner/XPathMatcher.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/ids/IdAttribute.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/ids/IdFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/ids/IdElement.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/ids/IdDocument.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/input/scanner/ElementListener.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/input/scanner/JakartaRegExpXPathMatcher.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2002-2014 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the

JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/ids/doc-files/TestIds.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2011 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
derived from this software without prior written permission. For
written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
may "JDOM" appear in their name, without prior
written permission
from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/LineSeparator.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/DefaultSAXHandlerFactory.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2000 Brett McLaughlin & Jason Hunter.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
 notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
 notice, this list of conditions, and the disclaimer that follows
 these conditions in the documentation and/or other materials
 provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
 derived from this software without prior written permission.  For
 written permission, please contact license@jdom.org.

4. Products derived from this software may not be called "JDOM", nor
 may "JDOM" appear in their name, without prior written permission
 from the
 JDOM Project Management (pm@jdom.org).

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
  "This product includes software developed by the
   JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY,
 WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally

created by Brett McLaughlin <brett@jdom.org> and
Jason Hunter <jhunter@jdom.org>.  For more information on the
JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/test/cases/TestAttribute.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/test/cases/TestElement.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2012 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without prior
 written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED

WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JNamespace.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JText.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JParent.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/XPathExpression.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/internal/SystemProperty.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/jaxen/JDOM2Navigator.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/util/XPathDiagnosticImpl.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/SlimJDOMFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JNode.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/XPathDiagnostic.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JProcessingInstruction.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/CloneBase.java
*

/opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/XPathBuilder.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/Wrapper.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/xpath/xalan/NamespacePointer.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JSimpleCharacterContent.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/jaxen/JaxenCompiled.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/stax/package-info.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JDocument.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JDOMImplementation.java
*
/opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JCDATA.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/jaxen/JDOMCoreNavigator.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/xpath/xalan/JDOM2DTM.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JDOMConfiguration.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JContent.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/test/cases/xpath/AbstractTestXPathCompiled.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/XPathFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JEntityRef.java
*
/opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JDocType.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JSimpleContent.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/util/AbstractXPathCompiled.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JNodeList.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/util/IteratorIterable.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/input/sax/XMLReaderJAXPFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/DOM.java

* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JAttribute.java
*
/opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JElement.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JComment.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/xpath/jaxen/JaxenXPathFactory.java
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/contrib/dom/JNamespaceAware.java
No license file was found, but licenses were detected in source scan.

/*--

Copyright (C) 2000-2011 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission.  For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without
 prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED

WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072629600_1595045934.09/0/1242-jdom2-2-0-6-sources-zip/org/jdom2/output/EscapeStrategy.java

# 1.45 retrofit 2.9.0

## 1.45.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/Retrofit.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/RequestBuilder.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/Callback.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/Converter.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2018 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/KotlinExtensions.kt
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/Invocation.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/internal/EverythingIsNonNull.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/Body.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/Url.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/OkHttpCall.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/BuiltInConverters.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/Response.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/OPTIONS.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/HttpServiceMethod.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/Call.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/DefaultCallAdapterFactory.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/ParameterHandler.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/ServiceMethod.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/RequestFactory.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/CallAdapter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/CompletableFutureCallAdapterFactory.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/HeaderMap.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/HttpException.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/HEAD.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/FormUrlEncoded.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/PUT.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/Multipart.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/Header.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/GET.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/Field.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-

jar/retrofit2/http/DELETE.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/QueryName.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/Query.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/Path.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/POST.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/Platform.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/HTTP.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/Headers.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/PATCH.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/Part.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/FieldMap.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/PartMap.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/Streaming.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/QueryMap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/SkipCallbackExecutorImpl.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/SkipCallbackExecutor.java
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/http/Tag.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/OptionalConverterFactory.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1083957721_1598378129.06/0/retrofit-2-9-0-sources-jar/retrofit2/Utils.java

# 1.46 swagger-core 1.6.2

## 1.46.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright 2020 SmartBear Software Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.47 swagger-annotations 1.6.2

## 1.47.1 Available under license :

No license file was found, but licenses were detected in source scan.

Bundle-License: http://www.apache.org/licenses/LICENSE-2.0.html

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1088518024_1599800722.99/0/swagger-annotations-1-6-2-jar/META-INF/MANIFEST.MF

# 1.48 netty-resolver 4.1.52.Final

## 1.48.1 Available under license :

No license file was found, but licenses were detected in source scan.

~ Copyright 2014 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~  http://www.apache.org/licenses/LICENSE
2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/META-INF/maven/io.netty/netty-resolver/pom.xml
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/RoundRobinInetAddressResolver.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/AddressResolverGroup.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/NameResolver.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/SimpleNameResolver.java
*
 /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/NoopAddressResolverGroup.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/NoopAddressResolver.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):

* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/HostsFileEntries.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/ResolvedAddressTypes.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/HostsFileEntriesResolver.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/AbstractAddressResolver.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/HostsFileParser.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/InetNameResolver.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/InetSocketAddressResolver.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/DefaultHostsFileEntriesResolver.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/DefaultAddressResolverGroup.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/CompositeNameResolver.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/AddressResolver.java
* /opt/ws_local/PERMITS_SQL/1093815881_1601014548.38/0/netty-resolver-4-1-52-final-sources-jar/io/netty/resolver/DefaultNameResolver.java

# 1.49 netty-codec 4.1.52.Final

## 1.49.1 Available under license :

No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~  http://www.apache.org/licenses/LICENSE
2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/META-INF/maven/io.netty/netty-codec/pom.xml
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Lz4XXHash32.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/MarshallingEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/ClassLoaderClassResolver.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/EncoderException.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/MarshallerProvider.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/MessageToMessageCodec.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/base64/Base64Encoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/JZlibDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/string/StringDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/CompatibleMarshallingDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/JZlibEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/CompactObjectInputStream.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/CompressionException.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/ZlibEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/DecoderResult.java

* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/SnappyFrameEncoder.java
*
/opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/ThreadLocalUnmarshallerProvider.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/MessageToMessageDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/FixedLengthFrameDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/JdkZlibEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/DecoderException.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/ZlibWrapper.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/Delimiters.java
*
/opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/bytes/ByteArrayDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/SoftReferenceMap.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Snappy.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/LimitingByteInput.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/ZlibUtil.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/base64/Base64Decoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/ChannelBufferByteOutput.java
*
/opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/ChannelBufferByteInput.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/ObjectEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/ZlibCodecFactory.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/ByteToMessageDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/ThreadLocalMarshallerProvider.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/package-info.java
*
/opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/ZlibDecoder.java

* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/PrematureChannelClosureException.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/UnmarshallerProvider.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/ObjectEncoderOutputStream.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/LineBasedFrameDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/SnappyFrameDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/bytes/ByteArrayEncoder.java
*
/opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/ReferenceMap.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/base64/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/WeakReferenceMap.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/xml/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/CompactObjectOutputStream.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/ContextBoundUnmarshallerProvider.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/MessageAggregator.java
*
/opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/ClassResolver.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/LengthFieldPrepender.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/ClassResolvers.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/TooLongFrameException.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/CorruptedFrameException.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/CodecException.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/package-info.java
*
/opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/DecompressionException.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/UnsupportedMessageTypeException.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-

jar/io/netty/handler/codec/MessageToByteEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/string/StringEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/protobuf/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/ObjectDecoder.java
*
/opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/ByteToMessageCodec.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/MarshallingDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/ReplayingDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/CompatibleMarshallingEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/CompatibleObjectEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/string/package-info.java
*
/opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/CachingClassResolver.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/ReplayingDecoderByteBuf.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/DefaultUnmarshallerProvider.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/DelimiterBasedFrameDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/serialization/ObjectDecoderInputStream.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/marshalling/DefaultMarshallerProvider.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/MessageToMessageEncoder.java
*
/opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/bytes/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:

*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/ValueConverter.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/CharSequenceValueConverter.java
*
 /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/DefaultHeadersImpl.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/*
* Written by Robert Harder and released to the public domain, as explained at
* http://creativecommons.org/licenses/publicdomain
*/
/**
* Enumeration of supported Base64 dialects.
* <p>
* The internal lookup tables in this class has been derived from
* <a href="http://iharder.sourceforge.net/current/java/base64/">Robert
 Harder's Public Domain
* Base64 Encoder/Decoder</a>.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/base64/Base64Dialect.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2Decoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/json/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2BitWriter.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2BitReader.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/SnappyFramedEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2Encoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/FastLz.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2BlockDecompressor.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Crc32.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2MTFAndRLE2StageEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/FastLzFrameDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/LzfEncoder.java

* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Lz4Constants.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2HuffmanStageDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/AsciiHeadersEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/MessageAggregationException.java
*
 /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/SnappyFramedDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2HuffmanStageEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Lz4FrameDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/json/JsonObjectDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/LzfDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2BlockCompressor.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/FastLzFrameEncoder.java
*
 /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2Constants.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2HuffmanAllocator.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Lz4FrameEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/LzmaFrameEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/DecoderResultProvider.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2DivSufSort.java
*
 /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2Rand.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Bzip2MoveToFrontTable.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance

* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/*
* Written by Robert Harder and released to the public domain, as explained at
* http://creativecommons.org/licenses/publicdomain
*/
/**
* Utility class for {@link ByteBuf} that encodes and decodes to and from
* <a href="http://en.wikipedia.org/wiki/Base64">Base64</a> notation.
* <p>
*
 The encoding and decoding algorithm in this class has been derived from
* <a href="http://iharder.sourceforge.net/current/java/base64/">Robert Harder's Public Domain
* Base64 Encoder/Decoder</a>.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/base64/Base64.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2013 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/JdkZlibDecoder.java

* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/Crc32c.java
*
 /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/xml/XmlFrameDecoder.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/string/LineSeparator.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/string/LineEncoder.java
*
 /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/CodecOutputList.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/DateFormatter.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/DatagramPacketDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/ByteBufChecksum.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/DatagramPacketEncoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/compression/CompressionUtil.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a

* copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/DefaultHeaders.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/EmptyHeaders.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/Headers.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/protobuf/ProtobufDecoderNano.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/UnsupportedValueConverter.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/protobuf/ProtobufVarint32LengthFieldPrepender.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/ProtocolDetectionState.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/protobuf/ProtobufVarint32FrameDecoder.java

* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/protobuf/ProtobufEncoderNano.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/ProtocolDetectionResult.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/protobuf/ProtobufDecoder.java
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/HeadersUtils.java
*
 /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/protobuf/ProtobufEncoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
* A decoder that splits the received {@link ByteBuf}s dynamically by the
* value of the length field in the message.  It is particularly useful when you
* decode a binary message which has an integer header field that represents the
* length of the message body or the whole message.
*
 <p>
* {@link LengthFieldBasedFrameDecoder} has many configuration parameters so
* that it can decode any message with a length field, which is often seen in
* proprietary client-server protocols. Here are some example that will give
* you the basic idea on which option does what.
*
* <h3>2 bytes length field at offset 0, do not strip header</h3>
*
* The value of the length field in this example is <tt>12 (0x0C)</tt> which
* represents the length of "HELLO, WORLD".  By default, the decoder assumes
* that the length field represents the number of the bytes that follows the
* length field.  Therefore, it can be decoded with the simplistic parameter
* combination.
* <pre>
```

```
* <b>lengthFieldOffset</b>   = <b>0</b>
* <b>lengthFieldLength</b>   = <b>2</b>
* lengthAdjustment    = 0
* initialBytesToStrip = 0 (= do not strip header)
*
* BEFORE DECODE (14 bytes)         AFTER DECODE (14 bytes)
* +--------+----------------+      +--------+----------------+
* | Length | Actual Content
 |----->| Length | Actual Content |
* | 0x000C | "HELLO, WORLD" |      | 0x000C | "HELLO, WORLD" |
* +--------+----------------+      +--------+----------------+
* </pre>
*
* <h3>2 bytes length field at offset 0, strip header</h3>
*
* Because we can get the length of the content by calling
* {@link ByteBuf#readableBytes()}, you might want to strip the length
* field by specifying <tt>initialBytesToStrip</tt>.  In this example, we
* specified <tt>2</tt>, that is same with the length of the length field, to
* strip the first two bytes.
* <pre>
* lengthFieldOffset   = 0
* lengthFieldLength   = 2
* lengthAdjustment    = 0
* <b>initialBytesToStrip</b> = <b>2</b> (= the length of the Length field)
*
* BEFORE DECODE (14 bytes)         AFTER DECODE (12 bytes)
* +--------+----------------+      +----------------+
* | Length | Actual Content |----->| Actual Content |
* | 0x000C | "HELLO, WORLD" |      | "HELLO, WORLD" |
* +--------+----------------+      +----------------+
 * </pre>
*
* <h3>2 bytes length field at offset 0, do not strip header, the length field
*     represents the length of the whole message</h3>
*
* In most cases, the length field represents the length of the message body
* only, as shown in the previous examples.  However, in some protocols, the
* length field represents the length of the whole message, including the
* message header.  In such a case, we specify a non-zero
* <tt>lengthAdjustment</tt>.  Because the length value in this example message
* is always greater than the body length by <tt>2</tt>, we specify <tt>-2</tt>
* as <tt>lengthAdjustment</tt> for compensation.
* <pre>
* lengthFieldOffset   =  0
* lengthFieldLength   =  2
* <b>lengthAdjustment</b>    = <b>-2</b> (= the length of the Length field)
* initialBytesToStrip =  0
```

```
*
* BEFORE DECODE (14 bytes)         AFTER DECODE (14 bytes)
* +--------+----------------+      +--------+----------------+
* | Length | Actual Content |----->| Length | Actual
 Content |
* | 0x000E | "HELLO, WORLD" |      | 0x000E | "HELLO, WORLD" |
* +--------+----------------+      +--------+----------------+
* </pre>
*
* <h3>3 bytes length field at the end of 5 bytes header, do not strip header</h3>
*
* The following message is a simple variation of the first example.  An extra
* header value is prepended to the message.  <tt>lengthAdjustment</tt> is zero
* again because the decoder always takes the length of the prepended data into
* account during frame length calculation.
* <pre>
* <b>lengthFieldOffset</b>   = <b>2</b> (= the length of Header 1)
* <b>lengthFieldLength</b>   = <b>3</b>
* lengthAdjustment    = 0
* initialBytesToStrip = 0
*
* BEFORE DECODE (17 bytes)                      AFTER DECODE (17 bytes)
* +----------+----------+----------------+      +----------+----------+----------------+
* | Header 1 |  Length  | Actual Content |----->| Header 1 |  Length  | Actual Content |
* |  0xCAFE  | 0x00000C | "HELLO, WORLD" |      |
  0xCAFE  | 0x00000C | "HELLO, WORLD" |
* +----------+----------+----------------+      +----------+----------+----------------+
* </pre>
*
* <h3>3 bytes length field at the beginning of 5 bytes header, do not strip header</h3>
*
* This is an advanced example that shows the case where there is an extra
* header between the length field and the message body.  You have to specify a
* positive <tt>lengthAdjustment</tt> so that the decoder counts the extra
* header into the frame length calculation.
* <pre>
* lengthFieldOffset   = 0
* lengthFieldLength   = 3
* <b>lengthAdjustment</b>    = <b>2</b> (= the length of Header 1)
* initialBytesToStrip = 0
*
* BEFORE DECODE (17 bytes)                      AFTER DECODE (17 bytes)
* +----------+----------+----------------+      +----------+----------+----------------+
* |  Length  | Header 1 | Actual Content |----->|  Length  | Header 1 | Actual Content |
* | 0x00000C |  0xCAFE  | "HELLO, WORLD" |      | 0x00000C |  0xCAFE  | "HELLO,
 WORLD" |
* +----------+----------+----------------+      +----------+----------+----------------+
* </pre>
```

```
*
* <h3>2 bytes length field at offset 1 in the middle of 4 bytes header,
*     strip the first header field and the length field</h3>
*
* This is a combination of all the examples above.  There are the prepended
* header before the length field and the extra header after the length field.
* The prepended header affects the <tt>lengthFieldOffset</tt> and the extra
* header affects the <tt>lengthAdjustment</tt>.  We also specified a non-zero
* <tt>initialBytesToStrip</tt> to strip the length field and the prepended
* header from the frame.  If you don't want to strip the prepended header, you
* could specify <tt>0</tt> for <tt>initialBytesToSkip</tt>.
* <pre>
* lengthFieldOffset   = 1 (= the length of HDR1)
* lengthFieldLength   = 2
* <b>lengthAdjustment</b>    = <b>1</b> (= the length of HDR2)
* <b>initialBytesToStrip</b> = <b>3</b> (= the length of HDR1 + LEN)
 *
* BEFORE DECODE (16 bytes)                    AFTER DECODE (13 bytes)
* +------+--------+------+----------------+      +------+----------------+
* | HDR1 | Length | HDR2 | Actual Content |----->| HDR2 | Actual Content |
* | 0xCA | 0x000C | 0xFE | "HELLO, WORLD" |      | 0xFE | "HELLO, WORLD" |
* +------+--------+------+----------------+      +------+----------------+
* </pre>
*
* <h3>2 bytes length field at offset 1 in the middle of 4 bytes header,
*     strip the first header field and the length field, the length field
*     represents the length of the whole message</h3>
*
* Let's give another twist to the previous example.  The only difference from
* the previous example is that the length field represents the length of the
* whole message instead of the message body, just like the third example.
* We have to count the length of HDR1 and Length into <tt>lengthAdjustment</tt>.
* Please note that we don't need to take the length of HDR2 into account
* because
 the length field already includes the whole header length.
* <pre>
* lengthFieldOffset   =  1
* lengthFieldLength   =  2
* <b>lengthAdjustment</b>    = <b>-3</b> (= the length of HDR1 + LEN, negative)
* <b>initialBytesToStrip</b> = <b> 3</b>
*
* BEFORE DECODE (16 bytes)                    AFTER DECODE (13 bytes)
* +------+--------+------+----------------+      +------+----------------+
* | HDR1 | Length | HDR2 | Actual Content |----->| HDR2 | Actual Content |
* | 0xCA | 0x0010 | 0xFE | "HELLO, WORLD" |      | 0xFE | "HELLO, WORLD" |
* +------+--------+------+----------------+      +------+----------------+
* </pre>
* @see LengthFieldPrepender
```

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815845_1601014627.54/0/netty-codec-4-1-52-final-sources-jar/io/netty/handler/codec/LengthFieldBasedFrameDecoder.java

# 1.50 netty-handler 4.1.52.Final

## 1.50.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SslCompletionEvent.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/JdkAlpnSslEngine.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OptionalSslHandler.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SslCloseCompletionEvent.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/JdkAlpnSslUtils.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ConscryptAlpnSslEngine.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ocsp/OcspClientHandler.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ocsp/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/Conscrypt.java

* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/AbstractSniHandler.java
*
 /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SslClientHelloHandler.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SniCompletionEvent.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ipfilter/UniqueIpFilter.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslEngine.java
*
 /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ipfilter/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/PemReader.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/JettyAlpnSslEngine.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/util/ThreadLocalInsecureRandom.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslSessionContext.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SslContext.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/JdkDefaultApplicationProtocolNegotiator.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslEngineMap.java
*

/opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ipfilter/RuleBasedIpFilter.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslDefaultApplicationProtocolNegotiator.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ipfilter/IpFilterRuleType.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SupportedCipherSuiteFilter.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SslUtils.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslApplicationProtocolNegotiator.java
*
/opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/CipherSuiteFilter.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslX509Certificate.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSsl.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/JdkApplicationProtocolNegotiator.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolConfig.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslNpnApplicationProtocolNegotiator.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslServerContext.java
*
/opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SniHandler.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/JdkBaseApplicationProtocolNegotiator.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ipfilter/IpFilterRule.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/JdkAlpnApplicationProtocolNegotiator.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ipfilter/IpSubnetFilterRule.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolNegotiator.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/JdkSslServerContext.java
*
/opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/JettyNpnSslEngine.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/CipherSuiteConverter.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-

jar/io/netty/handler/ssl/JdkSslEngine.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/Java7SslParametersUtils.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslSessionStats.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/util/OpenJdkSelfSignedCertGenerator.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/util/BouncyCastleSelfSignedCertGenerator.java
*
 /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolUtil.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/traffic/GlobalChannelTrafficCounter.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/util/FingerprintTrustManagerFactory.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SslProvider.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslServerSessionContext.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/JdkSslContext.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ipfilter/AbstractRemoteAddressFilter.java
*
 /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/util/SimpleTrustManagerFactory.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslClientContext.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/JdkNpnApplicationProtocolNegotiator.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/util/InsecureTrustManagerFactory.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/IdentityCipherSuiteFilter.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslContext.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/JdkSslClientContext.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version
* 2.0 (the "License"); you may not use this file except in compliance with the
* License. You may obtain a copy of the License at:
*

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/flow/FlowControlHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SslContextBuilder.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolAccessor.java
*
 /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolNames.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslJavaxX509Certificate.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ClientAuth.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslSessionTicketKey.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolNegotiationHandler.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
// Try the OpenJDK's proprietary implementation.
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/util/SelfSignedCertificate.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/logging/LogLevel.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SslHandler.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/stream/package-info.java
*
/opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/timeout/TimeoutException.java

* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/stream/ChunkedNioStream.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/stream/ChunkedInput.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/logging/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/timeout/IdleState.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/logging/LoggingHandler.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/stream/ChunkedWriteHandler.java
*
 /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/timeout/WriteTimeoutException.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/stream/ChunkedStream.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/stream/ChunkedFile.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/timeout/WriteTimeoutHandler.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/timeout/IdleStateEvent.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/NotSslRecordException.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/util/package-info.java
*
 /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/timeout/ReadTimeoutException.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/timeout/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/timeout/ReadTimeoutHandler.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/timeout/IdleStateHandler.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/stream/ChunkedNioFile.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:

*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/util/X509TrustManagerWrapper.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/DelegatingSslContext.java
*
 /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/flush/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/flow/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ReferenceCountedOpenSslServerContext.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ReferenceCountedOpenSslContext.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/PemValue.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/flush/FlushConsolidationHandler.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/PemEncoded.java
*
 /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/PemX509Certificate.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/Java8SslUtils.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ReferenceCountedOpenSslClientContext.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslKeyMaterialManager.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslCertificateException.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ReferenceCountedOpenSslEngine.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/PemPrivateKey.java
No license file was found, but licenses were detected in source scan.

# The Netty Project licenses this file to you under the Apache License,

# version 2.0 (the "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at:
#   http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/META-INF/native-image/io.netty/handler/native-image.properties
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2013 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SslHandshakeCompletionEvent.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):

* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SslMasterKeyHandler.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/KeyManagerFactoryWrapper.java
*
 /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/util/X509KeyManagerWrapper.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/PseudoRandomFunction.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/address/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslPrivateKeyMethod.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/address/DynamicAddressConnectHandler.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/util/SimpleKeyManagerFactory.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/TrustManagerFactoryWrapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2011 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/traffic/AbstractTrafficShapingHandler.java
No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
 ~
 ~ The Netty Project licenses this file to you under the Apache License,
 ~ version 2.0 (the "License"); you may not use this file except in compliance
 ~ with the License. You may obtain a copy of the License at:
 ~

~ http://www.apache.org/licenses/LICENSE
2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/META-INF/maven/io.netty/netty-handler/pom.xml
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2020 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/logging/ByteBufFormat.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/address/ResolveAddressHandler.java
*
/opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ipfilter/IpSubnetFilterRuleComparator.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ipfilter/IpSubnetFilter.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SslHandshakeTimeoutException.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,

* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/traffic/GlobalChannelTrafficShapingHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/traffic/GlobalTrafficShapingHandler.java
*
 /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/traffic/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2018 The Netty Project
*

* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslCachingX509KeyManagerFactory.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslCachingKeyMaterialProvider.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslSession.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslPrivateKey.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslTlsv13X509ExtendedTrustManager.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslKeyMaterialProvider.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/DefaultOpenSslKeyMaterial.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/ExtendedOpenSslSession.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslKeyMaterial.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/SignatureAlgorithmConverter.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslX509TrustManagerWrapper.java
* /opt/ws_local/PERMITS_SQL/1093815938_1601014541.84/0/netty-handler-4-1-52-final-sources-jar/io/netty/handler/ssl/OpenSslX509KeyManagerFactory.java

# 1.51 netty-transport 4.1.52.Final

## 1.51.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2018 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/SimpleUserEventChannelHandler.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/nio/NioChannelOption.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/ChannelOutputShutdownEvent.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DelegatingChannelPromiseNotifier.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/oio/OioDatagramChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/internal/ChannelUtils.java

* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/bootstrap/FailedChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/PendingBytesTracker.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/ChannelOutputShutdownException.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/internal/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/oio/DefaultOioDatagramChannelConfig.java
*
 /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/nio/SelectedSelectionKeySetSelector.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/ChannelInputShutdownReadComplete.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2020 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/StacklessClosedChannelException.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ExtendedClosedChannelException.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelHandlerMask.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/EventLoopTaskQueueFactory.java
No license file was found, but licenses were detected in source scan.

# The Netty Project licenses this file to you under the Apache License,
# version 2.0 (the "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at:
#   http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/META-INF/native-image/io.netty/transport/native-image.properties
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/embedded/EmbeddedChannelId.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ReflectiveChannelFactory.java

*
 /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelFactory.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/AbstractEventLoop.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/PendingWriteQueue.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2013 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/oio/OioByteStreamChannel.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
* Handles an I/O event or intercepts an I/O operation, and forwards it to its next handler in
* its {@link ChannelPipeline}.

```
*
* <h3>Sub-types</h3>
* <p>
* {@link ChannelHandler} itself does not provide many methods, but you usually have to implement one of its
subtypes:
* <ul>
* <li>{@link
 ChannelInboundHandler} to handle inbound I/O events, and</li>
* <li>{@link ChannelOutboundHandler} to handle outbound I/O operations.</li>
* </ul>
* </p>
* <p>
* Alternatively, the following adapter classes are provided for your convenience:
* <ul>
* <li>{@link ChannelInboundHandlerAdapter} to handle inbound I/O events,</li>
* <li>{@link ChannelOutboundHandlerAdapter} to handle outbound I/O operations, and</li>
* <li>{@link ChannelDuplexHandler} to handle both inbound and outbound events</li>
* </ul>
* </p>
* <p>
* For more information, please refer to the documentation of each subtype.
* </p>
*
* <h3>The context object</h3>
* <p>
* A {@link ChannelHandler} is provided with a {@link ChannelHandlerContext}
* object.  A {@link ChannelHandler} is supposed to interact with the
* {@link ChannelPipeline} it belongs to via a context object.  Using the
* context object, the {@link ChannelHandler} can pass events upstream or
* downstream, modify the pipeline dynamically,
 or store the information
* (using {@link AttributeKey}s) which is specific to the handler.
*
* <h3>State management</h3>
*
* A {@link ChannelHandler} often needs to store some stateful information.
* The simplest and recommended approach is to use member variables:
* <pre>
* public interface Message {
*     // your methods here
* }
*
* public class DataServerHandler extends {@link SimpleChannelInboundHandler}&lt;Message&gt; {
*
*     <b>private boolean loggedIn;</b>
*
*     {@code @Override}
*     public void channelRead0({@link ChannelHandlerContext} ctx, Message message) {
```

```
*        if (message instanceof LoginMessage) {
*            authenticate((LoginMessage) message);
*            <b>loggedIn = true;</b>
*        } else (message instanceof GetDataMessage) {
*            if (<b>loggedIn</b>) {
*                ctx.writeAndFlush(fetchSecret((GetDataMessage) message));
*            } else {
*                fail();
*            }
*        }
 *     }
*     ...
* }
* </pre>
* Because the handler instance has a state variable which is dedicated to
* one connection, you have to create a new handler instance for each new
* channel to avoid a race condition where a unauthenticated client can get
* the confidential information:
* <pre>
* // Create a new handler instance per channel.
* // See {@link ChannelInitializer#initChannel(Channel)}.
* public class DataServerInitializer extends {@link ChannelInitializer}&lt;{@link Channel}&gt; {
*     {@code @Override}
*     public void initChannel({@link Channel} channel) {
*         channel.pipeline().addLast("handler", <b>new DataServerHandler()</b>);
*     }
* }
*
* </pre>
*
* <h4>Using {@link AttributeKey}s</h4>
*
* Although it's recommended to use member variables to store the state of a
* handler, for some reason you might not want to create many handler instances.
* In such a case, you can use {@link AttributeKey}s which is provided by
* {@link ChannelHandlerContext}:
 * <pre>
* public interface Message {
*     // your methods here
* }
*
* {@code @Sharable}
* public class DataServerHandler extends {@link SimpleChannelInboundHandler}&lt;Message&gt; {
*     private final {@link AttributeKey}&lt;{@link Boolean}&gt; auth =
*         {@link AttributeKey#valueOf(String) AttributeKey.valueOf("auth")};
*
*     {@code @Override}
*     public void channelRead({@link ChannelHandlerContext} ctx, Message message) {
```

```
*        {@link Attribute}&lt;{@link Boolean}&gt; attr = ctx.attr(auth);
*        if (message instanceof LoginMessage) {
*            authenticate((LoginMessage) o);
*            <b>attr.set(true)</b>;
*        } else (message instanceof GetDataMessage) {
*            if (<b>Boolean.TRUE.equals(attr.get())</b>) {
*                ctx.writeAndFlush(fetchSecret((GetDataMessage) o));
*            } else {
*                fail();
*            }
*        }
*    }
*    ...
* }
* </pre>
* Now that the state
 of the handler is attached to the {@link ChannelHandlerContext}, you can add the
* same handler instance to different pipelines:
* <pre>
* public class DataServerInitializer extends {@link ChannelInitializer}&lt;{@link Channel}&gt; {
*
*    private static final DataServerHandler <b>SHARED</b> = new DataServerHandler();
*
*    {@code @Override}
*    public void initChannel({@link Channel} channel) {
*        channel.pipeline().addLast("handler", <b>SHARED</b>);
*    }
* }
* </pre>
*
*
* <h4>The {@code @Sharable} annotation</h4>
* <p>
* In the example above which used an {@link AttributeKey},
* you might have noticed the {@code @Sharable} annotation.
* <p>
* If a {@link ChannelHandler} is annotated with the {@code @Sharable}
* annotation, it means you can create an instance of the handler just once and
* add it to one or more {@link ChannelPipeline}s multiple times without
* a race condition.
* <p>
* If this annotation is not specified, you have to create
 a new handler
* instance every time you add it to a pipeline because it has unshared state
* such as member variables.
* <p>
* This annotation is provided for documentation purpose, just like
* <a href="http://www.javaconcurrencyinpractice.com/annotations/doc/">the JCIP annotations</a>.
```

```
*
* <h3>Additional resources worth reading</h3>
* <p>
* Please refer to the {@link ChannelHandler}, and
* {@link ChannelPipeline} to find out more about inbound and outbound operations,
* what fundamental differences they have, how they flow in a  pipeline,  and how to handle
* the operation in your application.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelHandler.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultChannelHandlerContext.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2013 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
```

* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/AbstractEventLoopGroup.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelOutboundBuffer.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/bootstrap/ChannelFactory.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelProgressiveFutureListener.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelId.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ConnectTimeoutException.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/group/ChannelMatchers.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/MessageSizeEstimator.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultChannelId.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/oio/DefaultOioSocketChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/group/ChannelMatcher.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/oio/OioSocketChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/SimpleChannelInboundHandler.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelHandlerAdapter.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/group/ChannelGroup.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/oio/DefaultOioServerSocketChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/oio/OioServerSocketChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultChannelProgressivePromise.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultAddressedEnvelope.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultMessageSizeEstimator.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-

jar/io/netty/channel/group/ChannelGroupException.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelProgressivePromise.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/AddressedEnvelope.java
*
/opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/group/DefaultChannelGroup.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelProgressiveFuture.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/nio/SelectedSelectionKeySet.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/pool/AbstractChannelPoolMap.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/pool/package-info.java
*
/opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/pool/ChannelPoolHandler.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/pool/ChannelPool.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/pool/ChannelPoolMap.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/pool/FixedChannelPool.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultMaxMessagesRecvByteBufAllocator.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/pool/ChannelHealthChecker.java

* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/MaxBytesRecvByteBufAllocator.java
*
 /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/pool/SimpleChannelPool.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/MaxMessagesRecvByteBufAllocator.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultMaxBytesRecvByteBufAllocator.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/pool/AbstractChannelPoolHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/FixedRecvByteBufAllocator.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelInitializer.java
*
 /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/bootstrap/AbstractBootstrap.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/DefaultServerSocketChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/local/LocalAddress.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/EventLoopException.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/CombinedChannelDuplexHandler.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-

jar/io/netty/channel/Channel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelOption.java
*
/opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/nio/AbstractNioChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ServerChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelFuture.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/local/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/DatagramChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/nio/AbstractNioByteChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/oio/AbstractOioByteChannel.java
*
/opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/EventLoopGroup.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/embedded/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/bootstrap/Bootstrap.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/VoidChannelPromise.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/nio/NioDatagramChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelInboundHandlerAdapter.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultFileRegion.java
*
/opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/DatagramChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/local/LocalServerChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/nio/NioDatagramChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/bootstrap/ServerBootstrap.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultChannelPromise.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/local/LocalEventLoopGroup.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/AbstractChannelHandlerContext.java

*
/opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelOutboundHandlerAdapter.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/nio/NioEventLoopGroup.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/oio/OioEventLoopGroup.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/DefaultDatagramChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/bootstrap/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/SocketChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/ServerSocketChannel.java
*
/opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/CompleteChannelFuture.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelPromise.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelPromiseNotifier.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/SucceededChannelFuture.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/nio/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/DatagramPacket.java
*
/opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/SingleThreadEventLoop.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/embedded/EmbeddedEventLoop.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/FileRegion.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelPipelineException.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/EventLoop.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ThreadPerChannelEventLoopGroup.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/nio/AbstractNioMessageChannel.java
*
/opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/oio/package-info.java

* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelFutureListener.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/group/DefaultChannelGroupFuture.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultEventLoop.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/oio/AbstractOioChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/nio/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/oio/OioSocketChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultChannelPipeline.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/MultithreadEventLoopGroup.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultEventLoopGroup.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/InternetProtocolFamily.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ThreadPerChannelEventLoop.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/group/ChannelGroupFuture.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/nio/ProtocolFamilyConverter.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/oio/OioServerSocketChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelDuplexHandler.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelPipeline.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/group/CombinedIterator.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/oio/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/local/LocalChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/FailedChannelFuture.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelFlushPromiseNotifier.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-

jar/io/netty/channel/group/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/AbstractServerChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/embedded/EmbeddedSocketAddress.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/RecvByteBufAllocator.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/DefaultSocketChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/ChannelInputShutdownEvent.java
*
/opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelInboundHandler.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/nio/NioServerSocketChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/SocketChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/group/ChannelGroupFutureListener.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/local/LocalChannelRegistry.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/AbstractChannel.java
*
/opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelMetadata.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelHandlerContext.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/nio/NioTask.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/nio/NioSocketChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelPromiseAggregator.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelException.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelOutboundHandler.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/ServerSocketChannelConfig.java
*
/opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/package-info.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/embedded/EmbeddedChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/AdaptiveRecvByteBufAllocator.java

* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/oio/AbstractOioMessageChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/nio/NioEventLoop.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/oio/OioDatagramChannel.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/WriteBufferWaterMark.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/bootstrap/ServerBootstrapConfig.java
*
 /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultSelectStrategyFactory.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/PreferHeapByteBufAllocator.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelOutboundInvoker.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/bootstrap/BootstrapConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/socket/DuplexChannel.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/ChannelInboundInvoker.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/bootstrap/AbstractBootstrapConfig.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/DefaultSelectStrategy.java
*

/opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/SelectStrategyFactory.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/SelectStrategy.java
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/group/VoidChannelGroupFuture.java
No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE
2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/META-INF/maven/io.netty/netty-transport/pom.xml
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/CoalescingBufferQueue.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2017 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815931_1601014618.02/0/netty-transport-4-1-52-final-sources-jar/io/netty/channel/AbstractCoalescingBufferQueue.java

# 1.52 netty-buffer 4.1.52.Final

## 1.52.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
~ Copyright 2012 The Netty Project
 ~
 ~ The Netty Project licenses this file to you under the Apache License,
 ~ version 2.0 (the "License"); you may not use this file except in compliance
 ~ with the License. You may obtain a copy of the License at:
 ~
 ~   http://www.apache.org/licenses/LICENSE
2.0
 ~
 ~ Unless required by applicable law or agreed to in writing, software
 ~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 ~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 ~ License for the specific language governing permissions and limitations
 ~ under the License.
```

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815874_1601014588.69/0/netty-buffer-4-1-52-final-jar/META-INF/maven/io.netty/netty-buffer/pom.xml
No license file was found, but licenses were detected in source scan.

Bundle-License: http://www.apache.org/licenses/LICENSE-2.0

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815874_1601014588.69/0/netty-buffer-4-1-52-final-jar/META-INF/MANIFEST.MF
No license file was found, but licenses were detected in source scan.

# The Netty Project licenses this file to you under the Apache License,
# version 2.0 (the "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at:
#   http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1093815874_1601014588.69/0/netty-buffer-4-1-52-final-jar/META-INF/native-image/io.netty/buffer/native-image.properties

# 1.53 checker-qual 3.5.0

## 1.53.1 Available under license :

Checker Framework qualifiers
Copyright 2004-present by the Checker Framework developers

MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.54 netty-handler-proxy 4.1.52.Final

## 1.54.1 Available under license :

No license file was found, but licenses were detected in source scan.

~ Copyright 2014 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~  http://www.apache.org/licenses/LICENSE
2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
* /opt/cola/permits/1099818518_1602573656.03/0/netty-handler-proxy-4-1-52-final-sources-jar/META-INF/maven/io.netty/netty-handler-proxy/pom.xml
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1099818518_1602573656.03/0/netty-handler-proxy-4-1-52-final-sources-jar/io/netty/handler/proxy/Socks4ProxyHandler.java
* /opt/cola/permits/1099818518_1602573656.03/0/netty-handler-proxy-4-1-52-final-sources-jar/io/netty/handler/proxy/ProxyConnectException.java
*

/opt/cola/permits/1099818518_1602573656.03/0/netty-handler-proxy-4-1-52-final-sources-jar/io/netty/handler/proxy/ProxyHandler.java
* /opt/cola/permits/1099818518_1602573656.03/0/netty-handler-proxy-4-1-52-final-sources-jar/io/netty/handler/proxy/HttpProxyHandler.java
* /opt/cola/permits/1099818518_1602573656.03/0/netty-handler-proxy-4-1-52-final-sources-jar/io/netty/handler/proxy/package-info.java
* /opt/cola/permits/1099818518_1602573656.03/0/netty-handler-proxy-4-1-52-final-sources-jar/io/netty/handler/proxy/ProxyConnectionEvent.java
* /opt/cola/permits/1099818518_1602573656.03/0/netty-handler-proxy-4-1-52-final-sources-jar/io/netty/handler/proxy/Socks5ProxyHandler.java

# 1.55 netty-codec-socks 4.1.52.Final

## 1.55.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2013 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5AddressType.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksResponseType.java
*
/opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/SocksVersion.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksAuthStatus.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5CommandType.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5CommandStatus.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-

jar/io/netty/handler/codec/socksx/v5/Socks5AuthMethod.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksSubnegotiationVersion.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthStatus.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksProtocolVersion.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksCmdType.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksRequestType.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksAddressType.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksMessageType.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksAuthScheme.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksCmdStatus.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5AddressEncoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5AddressDecoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/SocksPortUnificationServerHandler.java
No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE
2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/META-INF/maven/io.netty/netty-codec-socks/pom.xml
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5InitialResponse.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5CommandRequest.java
*
/opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/UnknownSocksResponse.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksAuthResponseDecoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4CommandStatus.java

* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5CommandRequest.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5CommandResponse.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4CommandRequest.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/UnknownSocksRequest.java
*
/opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5PasswordAuthRequest.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/SocksMessage.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5PasswordAuthResponse.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthRequest.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4CommandType.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksCommonUtils.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksMessageEncoder.java
*
/opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5InitialRequest.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksInitRequestDecoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksInitResponseDecoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksAuthRequestDecoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksRequest.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksResponse.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4ClientDecoder.java
*
/opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthResponse.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4CommandResponse.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksInitRequest.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksCmdRequestDecoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-

jar/io/netty/handler/codec/socksx/v5/package-info.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksCmdResponse.java
*
 /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5InitialResponse.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksCmdResponseDecoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4ServerDecoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5InitialRequest.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksMessage.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v4/DefaultSocks4CommandRequest.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v4/DefaultSocks4CommandResponse.java
*
 /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksCmdRequest.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksAuthResponse.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5CommandResponse.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksAuthRequest.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/SocksInitResponse.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socks/package-info.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v4/AbstractSocks4Message.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5InitialResponseDecoder.java
*
/opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5Message.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4ClientEncoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v4/package-info.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5CommandResponseDecoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/AbstractSocksMessage.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/AbstractSocks5Message.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthRequestDecoder.java
*
/opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4ServerEncoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5CommandRequestDecoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5ServerEncoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthResponseDecoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5InitialRequestDecoder.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/package-info.java
* /opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5ClientEncoder.java
*
/opt/cola/permits/1099818508_1602573654.35/0/netty-codec-socks-4-1-52-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4Message.java

# 1.56 jna-platform 5.6.0

## 1.56.1 Available under license :

No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2010 Daniel Doubrovkine, All Rights Reserved
*

* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WinReg.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/DsGetDC.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Crypt32.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WinDef.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Secur32Util.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/BaseTSD.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Secur32.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Wdm.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/LMAccess.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/NtDll.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Advapi32Util.java
*

/opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Netapi32.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WinspoolUtil.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Win32Exception.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Ole32Util.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WinBase.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WinGDI.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WinUser.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/ShellAPI.java
*
/opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/SspiUtil.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WinCrypt.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/LMJoin.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/ShlObj.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/NTSecApi.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Ole32.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/NtDllUtil.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/LMCons.java
*
/opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Netapi32Util.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Winspool.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Guid.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Crypt32Util.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2015 Michael Freeman, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and

```
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/
```

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WininetUtil.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/VersionUtil.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/GDI32Util.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Wininet.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2019 Keve Müller
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0.
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
```

*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Win32VK.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2019 Daniel Widdis, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/
// version.

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/VersionHelpers.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2015 Andreas "PAX" L\u00FCck, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and

* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Psapi.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2014 Reinhard Pointner, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/mac/XAttr.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/mac/XAttrUtil.java
No license file was found, but licenses were detected in source scan.

```
/* Copyright (c) 2018 Matthias Bläsing, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/
```

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WinCryptUtil.java
No license file was found, but licenses were detected in source scan.

```
/* Copyright (c) 2007 Timothy Wall, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
```

* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/NTStatus.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/FileMonitor.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/FileUtils.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/W32FileMonitor.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/W32FileUtils.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/dnd/DropTargetPainter.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/dnd/DragHandler.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/dnd/GhostedDragImage.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/dnd/DropHandler.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/KeyboardUtils.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WinNT.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/GDI32.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/LMErr.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Wincon.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/unix/X11.java
No license file was found, but licenses were detected in source scan.

```
/* Copyright (c) 2010 Daniel Doubrovkine, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/
/**
   * Supports privacy (confidentiality)
   */
```

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Sspi.java
No license file was found, but licenses were detected in source scan.

```
/* Copyright (c) 2010 EugineLev, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
```

* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/W32Service.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Winsvc.java
*
 /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/W32ServiceManager.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2013 Tobias Wolf, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbPropertyPut.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbCmdlineArgs.java
*

/opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbAbstractMethod.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbDispInterface.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/COMInvoker.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbFunctionVTable.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbPropertyPutStub.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/TypeLibUtil.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/TlbImp.java
*
/opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbBase.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbFunctionStub.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbFunctionDispId.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbParameterNotFoundException.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbConst.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/COMUtils.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbEnum.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbInterface.java
*
/opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbPropertyGetStub.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbCoClass.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/TypeInfoUtil.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/tlb/imp/TlbPropertyGet.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2018 Václav Haisman, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0.

*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/linux/XAttr.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/linux/XAttrUtil.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2016 Adam Marcionek, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".

*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Ntifs.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WinioctlUtil.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2012 Tobias Wolf, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IRecordInfo.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/COMEarlyBindingObject.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/OleAuto.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/Unknown.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/ITypeLib.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Wtsapi32.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-

jar/com/sun/jna/platform/win32/COM/IDispatch.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/TypeComp.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/COMException.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/Dispatch.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/COMLateBindingObject.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/ITypeInfo.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/DBT.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IUnknown.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/COMBindingBaseObject.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/ITypeComp.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/TypeLib.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/TypeInfo.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/RecordInfo.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2019 Daniel Widdis, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/

```
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/
```

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/PowrProf.java
No license file was found, but licenses were detected in source scan.

```
/* Copyright (c) 2010 Daniel Doubrovkine, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/
/**
   * <code>
   BOOL WINAPI CreateProcessWithLogonW(
   _In_        LPCWSTR lpUsername,
   _In_opt_    LPCWSTR lpDomain,
   _In_        LPCWSTR lpPassword,
   _In_        DWORD dwLogonFlags,
   _In_opt_    LPCWSTR lpApplicationName,
   _Inout_opt_ LPWSTR lpCommandLine,
   _In_        DWORD dwCreationFlags,
   _In_opt_    LPVOID lpEnvironment,
   _In_opt_    LPCWSTR lpCurrentDirectory,
   _In_        LPSTARTUPINFOW lpStartupInfo,
   _Out_       LPPROCESS_INFORMATION lpProcessInfo
   );
```

```
 * </code>
 *
 * @param lpUsername
 *        [in]<br>
 *        The name of the user. This is the name of the user account to
 *        log on to. <br>
 *        If you use the UPN format, user@DNS_domain_name, the lpDomain
 *        parameter must be NULL.<br>
 *        The user account must have the Log On Locally permission on
 *        the local computer. <br>
 *        This permission is granted to all users on workstations and
 *        servers, but only to administrators on domain controllers.
 * @param lpDomain
 *        [in, optional]<br>
 *
The name of the domain or server whose account database
 *        contains the lpUsername account. <br>
 *        If this parameter is NULL, the user name must be specified in
 *        UPN format.
 * @param lpPassword
 *        [in]<br>
 *        The clear-text password for the lpUsername account.
 * @param dwLogonFlags
 *        [in]<br>
 *        The logon option. This parameter can be 0 (zero) or one of the
 *        following values. <br>
 *        LOGON_WITH_PROFILE: 0x00000001<br>
 *        Log on, then load the user profile in the HKEY_USERS registry
 *        key.<br>
 *        The function returns after the profile is loaded. <br>
 *        Loading the profile can be time-consuming, so it is best to
 *        use this value only if you must access the information in the
 *        HKEY_CURRENT_USER registry key.<br>
 *        Windows Server 2003:
The profile is unloaded after the new
 *        process is terminated, whether or not it has created child
 *        processes.<br>
 *        Windows XP: The profile is unloaded after the new process and
 *        all child processes it has created are terminated.<br>
 *        <br>
 *        LOGON_NETCREDENTIALS_ONLY: 0x00000002<br>
 *        Log on, but use the specified credentials on the network only.
 *        <br>
 *        The new process uses the same token as the caller, but the
 *        system creates a new logon session within LSA, and the process
 *        uses the specified credentials as the default credentials.
 *        <br>
 *        This value can be used to create a process that uses a
```

```
 *          different set of credentials locally than it does remotely.
 *          <br>
 *          This is useful in inter-domain scenarios where there is no
  *          trust relationship.<br>
 *          The system does not validate the specified credentials.<br>
 *          Therefore, the process can start, but it may not have access
 *          to network resources.
 * @param lpApplicationName
 *          [in, optional]<br>
 *          The name of the module to be executed. This module can be a
 *          Windows-based application.<br>
 *          It can be some other type of module (for example, MS-DOS or
 *          OS/2) if the appropriate subsystem is available on the local
 *          computer. The string can specify the full path and file name
 *          of the module to execute or it can specify a partial name.
 *          <br>
 *          If it is a partial name, the function uses the current drive
 *          and current directory to complete the specification. <br>
 *          The function does not use the search path. This parameter
must
 *          include the file name extension; no default extension is
 *          assumed. The lpApplicationName parameter can be NULL, and the
 *          module name must be the first white space-delimited token in
 *          the lpCommandLine string.<br>
 *          If you are using a long file name that contains a space, use
 *          quoted strings to indicate where the file name ends and the
 *          arguments begin; otherwise, the file name is ambiguous. For
 *          example, the following string can be interpreted in different
 *          ways:<br>
 *          "c:\program files\sub dir\program name"<br>
 *          The system tries to interpret the possibilities in the
 *          following order:<br>
 *          <br>
 *          c:\program.exe files\sub dir\program name<br>
 *          c:\program files\sub.exe dir\program name<br>
 *          c:\program files\sub dir\program.exe
name<br>
 *          c:\program files\sub dir\program name.exe<br>
 *          <br>
 *          If the executable module is a 16-bit application,
 *          lpApplicationName should be NULL, and the string pointed to by
 *          lpCommandLine should specify the executable module and its
 *          arguments.
 * @param lpCommandLine
 *          [in, out, optional]<br>
 *          The command line to be executed. The maximum length of this
 *          string is 1024 characters. <br>
 *          If lpApplicationName is NULL, the module name portion of
```

```
 *      lpCommandLine is limited to MAX_PATH characters.<br>
 *      The function can modify the contents of this string. <br>
 *      Therefore, this parameter cannot be a pointer to read-only
 *      memory (such as a const variable or a literal string).<br>
 *      If this parameter is a constant string, the
function may cause
 *      an access violation.<br>
 *      The lpCommandLine parameter can be NULL, and the function uses
 *      the string pointed to by lpApplicationNameas the command line.
 *      <br>
 *      <br>
 *      If both lpApplicationName and lpCommandLine are non-NULL,
 *      *lpApplicationName specifies the module to execute, and
 *      *lpCommandLine specifies the command line.<br>
 *      The new process can use GetCommandLine to retrieve the entire
 *      command line.<br>
 *      Console processes written in C can use the argc and argv
 *      arguments to parse the command line. <br>
 *      Because argv[0] is the module name, C programmers typically
 *      repeat the module name as the first token in the command line.
 *      <br>
 *      If lpApplicationName is NULL, the first white space-delimited

 *      token of the command line specifies the module name.<br>
 *      If you are using a long file name that contains a space, use
 *      quoted strings to indicate where the file name ends and the
 *      arguments begin (see the explanation for the lpApplicationName
 *      parameter). If the file name does not contain an extension,
 *      .exe is appended. Therefore, if the file name extension is
 *      .com, this parameter must include the .com extension. If the
 *      file name ends in a period with no extension, or if the file
 *      name contains a path, .exe is not appended. If the file name
 *      does not contain a directory path, the system searches for the
 *      executable file in the following sequence:<br>
 *      <br>
 *      1. The directory from which the application loaded.<br>
 *      2. The current directory for the parent
process.<br>
 *      3. The 32-bit Windows system directory. Use the
 *      GetSystemDirectory function to get the path of this directory.
 *      <br>
 *      4. The 16-bit Windows system directory. There is no function
 *      that obtains the path of this directory, but it is searched.
 *      <br>
 *      5. The Windows directory. Use the GetWindowsDirectory function
 *      to get the path of this directory.<br>
 *      6. The directories that are listed in the PATH environment
 *      variable. Note that this function does not search the
```

```
 *          per-application path specified by the App Paths registry key.
 *          To include this per-application path in the search sequence,
 *          use the ShellExecute function.<br>
 *          <br>
 *          The system adds a null character to the command line string to
 *          separate the
file name from the arguments. This divides the
 *          original string into two strings for internal processing.<br>
 * @param dwCreationFlags
 *          The flags that control how the process is created. <br>
 *          The CREATE_DEFAULT_ERROR_MODE, CREATE_NEW_CONSOLE, and
 *          CREATE_NEW_PROCESS_GROUP flags are enabled by default. <br>
 *          Even if you do not set the flag, the system functions as if it
 *          were set. You can specify additional flags as noted.<br>
 *          <br>
 *          CREATE_DEFAULT_ERROR_MODE: 0x04000000<br>
 *          The new process does not inherit the error mode of the calling
 *          process. <br>
 *          Instead, CreateProcessWithLogonW gives the new process the
 *          current default error mode. <br>
 *          An application sets the current default error mode by calling
 *          SetErrorMode. This flag is enabled by
default.<br>
 *          <br>
 *          CREATE_NEW_CONSOLE: 0x00000010<br>
 *          The new process has a new console, instead of inheriting the
 *          parent's console. This flag cannot be used with the
 *          DETACHED_PROCESS flag.<br>
 *          This flag is enabled by default.<br>
 *          <br>
 *          CREATE_NEW_PROCESS_GROUP: 0x00000200<br>
 *          The new process is the root process of a new process group.
 *          <br>
 *          The process group includes all processes that are descendants
 *          of this root process.<br>
 *          The process identifier of the new process group is the same as
 *          the process identifier, which is returned in the lpProcessInfo
 *          parameter.<br>
 *          Process groups are used by the GenerateConsoleCtrlEvent
 *          function to enable sending a CTRL+C or CTRL+BREAK signal to a
 *
        group of console processes.<br>
 *          This flag is enabled by default.<br>
 *          <br>
 *          CREATE_SEPARATE_WOW_VDM: 0x00000800<br>
 *          This flag is only valid starting a 16-bit Windows-based
 *          application.<br>
 *          If set, the new process runs in a private Virtual DOS Machine
```

```
  *         (VDM).<br>
  *         By default, all 16-bit Windows-based applications run in a
  *         single, shared VDM.<br>
  *         The advantage of running separately is that a crash only
  *         terminates the single VDM; any other programs running in
  *         distinct VDMs continue to function normally.<br>
  *         Also, 16-bit Windows-based applications that run in separate
  *         VDMs have separate input queues, which means that if one
  *         application stops responding momentarily, applications in
  *         separate VDMs continue
to receive input. <br>
  *         CREATE_SUSPENDED: 0x00000004<br>
  *         The primary thread of the new process is created in a
  *         suspended state, and does not run until the ResumeThread
  *         function is called.<br>
  *         <br>
  *         CREATE_UNICODE_ENVIRONMENT: 0x00000400<br>
  *         Indicates the format of the lpEnvironment parameter.<br>
  *         If this flag is set, the environment block pointed to by
  *         lpEnvironment uses Unicode characters. <br>
  *         Otherwise, the environment block uses ANSI characters. <br>
  *         EXTENDED_STARTUPINFO_PRESENT: 0x00080000<br>
  *         The process is created with extended startup information; the
  *         lpStartupInfo parameter specifies a STARTUPINFOEX structure.
  *         <br>
  *         Windows Server 2003 and Windows XP: This value is not
  *         supported.<br>
  * @param
lpEnvironment
  *         [in, optional]<br>
  *         A pointer to an environment block for the new process.<br>
  *         If this parameter is NULL, the new process uses an environment
  *         created from the profile of the user specified by lpUsername.
  *         An environment block consists of a null-terminated block of
  *         null-terminated strings.<br>
  *         Each string is in the following form:<br>
  *         name=value<br>
  *         Because the equal sign (=) is used as a separator, it must not
  *         be used in the name of an environment variable.<br>
  *         An environment block can contain Unicode or ANSI characters.
  *         <br>
  *         If the environment block pointed to by lpEnvironment contains
  *         Unicode characters, ensure that dwCreationFlags includes
  *         CREATE_UNICODE_ENVIRONMENT.<br>
  *         If this parameter
is NULL and the environment block of the
  *         parent process contains Unicode characters, you must also
  *         ensure that dwCreationFlags includes
```

```
 *          CREATE_UNICODE_ENVIRONMENT.<br>
 *          An ANSI environment block is terminated by two 0 (zero) bytes:
 *          one for the last string and one more to terminate the block.
 *          <br>
 *          A Unicode environment block is terminated by four zero bytes:
 *          two for the last string and two more to terminate the block.
 *          <br>
 *          To retrieve a copy of the environment block for a specific
 *          user, use the CreateEnvironmentBlock function.<br>
 * @param lpCurrentDirectory
 *          [in, optional]<br>
 *          The full path to the current directory for the process.<br>
 *          The string can also specify a UNC path.<br>
 *          If this parameter is NULL, the new
process has the same
 *          current drive and directory as the calling process.<br>
 *          This feature is provided primarily for shells that need to
 *          start an application, and specify its initial drive and
 *          working directory.<br>
 * @param lpStartupInfo
 *          [in]<br>
 *          A pointer to a STARTUPINFO or STARTUPINFOEX structure. <br>
 *          The application must add permission for the specified user
 *          account to the specified window station and desktop, even for
 *          WinSta0\Default.<br>
 *          If the lpDesktop member is NULL or an empty string, the new
 *          process inherits the desktop and window station of its parent
 *          process.<br>
 *          The application must add permission for the specified user
 *          account to the inherited window station and desktop.<br>
 *          Windows XP: CreateProcessWithLogonW
adds permission for the
 *          specified user account to the inherited window station and
 *          desktop.<br>
 *          Handles in STARTUPINFO or STARTUPINFOEX must be closed with
 *          CloseHandle when they are no longer needed.<br>
 *          Important If the dwFlags member of the STARTUPINFO structure
 *          specifies STARTF_USESTDHANDLES, the standard handle fields are
 *          copied unchanged to the child process without validation.<br>
 *          The caller is responsible for ensuring that these fields
 *          contain valid handle values. Incorrect values can cause the
 *          child process to misbehave or crash.<br>
 *          Use the Application Verifier runtime verification tool to
 *          detect invalid handles.
 * @param lpProcessInfo
 *          [out]<br>
 *          A pointer to a PROCESS_INFORMATION structure that receives
 *
```

```
        identification information for the new process, including a
   *          handle to the process.<br>
   *          Handles in PROCESS_INFORMATION must be closed with the
   *          CloseHandle function when they are not needed.<br>
   * @return If the function succeeds, the return value is nonzero.<br>
   *        If the function fails, the return value is 0 (zero).<br>
   *        To get extended error information, call GetLastError.<br>
   *        Note that the function returns before the process has finished
   *        initialization.<br>
   *        If a required DLL cannot be located or fails to initialize, the
   *        process is terminated.<br>
   *        To get the termination status of a process, call
   *        GetExitCodeProcess.
   * @see <a href="http://msdn.microsoft.com/en-
us/library/windows/desktop/ms682431%28v=vs.85%29.aspx">MSDN</a>
   */
```

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Advapi32.java
No license file was found, but licenses were detected in source scan.

```
/* Copyright (c) 2018 Daniel Widdis, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/
```

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-

jar/com/sun/jna/platform/win32/PdhUtil.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/Wbemcli.java
*
 /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Cfgmgr32Util.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/WbemcliUtil.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Cfgmgr32.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2016 Minoru Sakamoto, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WevtapiUtil.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Winevt.java
*
 /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Wevtapi.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2011 Daniel Doubrovkine, All Rights Reserved
*

* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/
// MessageId: ERROR_DS_CONFIDENTIALITY_REQUIRED
// version.

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WinError.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2010 Timothy Wall, All Rights Reserved
* Copyright 2010 Digital Rapids Corp.
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*

* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WTypes.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/ObjBase.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2015 Andreas "PAX" L\u00FCck, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/DesktopWindow.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2007-2008 Timothy Wall, All Rights Reserved
* Copyright (c) 2007 Olivier Chafik
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and

* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/WindowUtils.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2017 Matthias Bläsing, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IComEnumVariantIterator.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/EnumVariant.java
*
 /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Winsock2.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/ShTypes.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IEnumVariant.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/COMInvokeException.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2015 Adam Marcionek, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Winnetwk.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/LMShare.java
*

/opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Mpr.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/AccCtrl.java
No license file was found, but licenses were detected in source scan.

```
/* Copyright (c) 2020 Torbjörn Svensson, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/
```

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/PsapiUtil.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2013 Ralf Hamberger, Markus Karg, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
```

*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/User32Util.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2010,2011 Daniel Doubrovkine, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/W32Errors.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2007-2013 Timothy Wall, All Rights Reserved
*

* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/mac/MacFileUtils.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2015 Daniel Widdis
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*

* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/mac/SystemB.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2017 Daniel Widdis, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/unix/solaris/LibKstat.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/linux/LibC.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2007, 2013 Timothy Wall, Markus Karg, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.

*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Shell32.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/User32.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Kernel32.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2019 Daniel Widdis
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package

* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/mac/IOReturnException.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/mac/CoreFoundation.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/mac/IOKit.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/mac/IOKitUtil.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/mac/DiskArbitration.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2015 Goldstein Lyor, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WinPerf.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/PdhMsg.java
*

/opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/unix/Resource.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/unix/LibC.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/unix/Reboot.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/unix/LibCAPI.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Pdh.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2014 Dr David H. Akehurst (itemis), All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/IConnectionPoint.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/ComEventCallbackCookie.java
*
/opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/AbstractComEventCallbackListener.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/UnknownListener.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/Moniker.java

* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/Factory.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/IComEnum.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/UnknownVTable.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/ComThread.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/IComEventCallbackListener.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/IDispatch.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/ObjectFactory.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/RunningObjectTable.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IPersistStream.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IPersist.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IConnectionPointContainer.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/annotation/ComProperty.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IDispatchCallback.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/IUnknown.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IRunningObjectTable.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/ProxyObject.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/ConnectionPointContainer.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IConnectionPoint.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/IRunningObjectTable.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/EnumMoniker.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IMoniker.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/RunningObjectTable.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-

jar/com/sun/jna/platform/win32/COM/util/IConnectionPointContainer.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/Convert.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/annotation/ComEventCallback.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/DispatchListener.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IEnumMoniker.java
*
 /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/annotation/ComMethod.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IUnknownCallback.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/annotation/ComInterface.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/ConnectionPoint.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/EnumMoniker.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/IRawDispatchHandle.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/DispatchVTable.java
*
 /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/CallbackProxy.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/annotation/ComObject.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/util/IComEventCallbackCookie.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IStream.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2020 Daniel Widdis, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*

* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/linux/ErrNo.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/unix/LibCUtil.java
*
 /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/linux/LibRT.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/linux/Fcntl.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/linux/Udev.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/linux/Mman.java
No license file was found, but licenses were detected in source scan.

/*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".

*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/KnownFolders.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/VerRsrc.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Winioctl.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/OaIdlUtil.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IEnumIDList.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/SetupApi.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Msi.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Ddeml.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Tlhelp32.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Shlwapi.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Version.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/DdemlUtil.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/OaIdl.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/COM/IShellFolder.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Variant.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 Martin Steiger
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:

*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/EnumConverter.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/PhysicalMonitorEnumerationAPI.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/FlagEnum.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Dxva2.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/HighLevelMonitorConfigurationAPI.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/LowLevelMonitorConfigurationAPI.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/EnumUtils.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2010, 2013 Daniel Doubrovkine, Markus Karg, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*

* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Shell32Util.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Kernel32Util.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2018,2020 Daniel Widdis, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/IPHlpAPI.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2011 Timothy Wall, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and

* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/WinRas.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Rasapi32Util.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/OpenGL32Util.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/wince/CoreDLL.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Rasapi32.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/OpenGL32.java
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/GL.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2011 Denis Tulskiy
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*

* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/mac/Carbon.java
No license file was found, but licenses were detected in source scan.

/* Copyright (c) 2018 Roshan Muralidharan, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/win32/Cryptui.java

No license file was found, but licenses were detected in source scan.

```
/* Copyright (c) 2007 Olivier Chafik, All Rights Reserved
* Copyright (c) 2008 Timothy Wall, All Rights Reserved
*
* The contents of this file is dual-licensed under 2
* alternative Open Source/Free licenses: LGPL 2.1 or later and
* Apache License 2.0. (starting with JNA version 4.0.0).
*
* You can freely decide which license you want to apply to
* the project.
*
* You may obtain a copy of the LGPL License at:
*
* http://www.gnu.org/licenses/licenses.html
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "LGPL2.1".
*
* You may obtain a copy of the Apache License at:
*
* http://www.apache.org/licenses/
*
* A copy is also included in the downloadable source code package
* containing JNA, in file "AL2.0".
*/
```

Found in path(s):
* /opt/cola/permits/1102347507_1603692766.03/0/jna-platform-5-6-0-sources-jar/com/sun/jna/platform/RasterRangesUtils.java

# 1.57 jna 5.6.0

## 1.57.1 Available under license :

Java Native Access project (JNA) is dual-licensed under 2
alternative Open Source/Free licenses: LGPL 2.1 or later and
Apache License 2.0. (starting with JNA version 4.0.0).

You can freely decide which license you want to apply to
the project.

You may obtain a copy of the LGPL License at:

http://www.gnu.org/licenses/licenses.html

A copy is also included in the downloadable source code package
containing JNA, in file "LGPL2.1", under the same directory

as this file.

You may obtain a copy of the Apache License at:

http://www.apache.org/licenses/

A copy is also included in the downloadable source code package
containing JNA, in file "AL2.0", under the same directory
as this file.

# 1.58 annotations 20.1.0

## 1.58.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2000-2020 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/Contract.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/ApiStatus.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/intellij/lang/annotations/Subst.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/Nls.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/VisibleForTesting.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/NonNls.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/intellij/lang/annotations/JdkConstants.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-

jar/org/jetbrains/annotations/Range.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/PropertyKey.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/NotNull.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/intellij/lang/annotations/MagicConstant.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/intellij/lang/annotations/RegExp.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/Async.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/TestOnly.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/MustBeInvokedByOverriders.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/intellij/lang/annotations/Flow.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/intellij/lang/annotations/Identifier.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/intellij/lang/annotations/Language.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/UnmodifiableView.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/Unmodifiable.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/Nullable.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/jetbrains/annotations/Debug.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/intellij/lang/annotations/PrintFormat.java
* /opt/cola/permits/1103213823_1604039616.52/0/annotations-20-1-0-sources-jar/org/intellij/lang/annotations/Pattern.java

# 1.59 httpclient 4.5.13

## 1.59.1 Available under license :

Apache HttpClient
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License

Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.60 joda-time 2.10.8

## 1.60.1 Available under license :

```
=============================================================================
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
=============================================================================
```

This product includes software developed by
Joda.org (https://www.joda.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or

(iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own
copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names
of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law
(such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.61 xerces-impl 2.12.0

## 1.61.1 Available under license :

No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE
2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-
jar/org/apache/xerces/xs/datatypes/package.html
No license file was found, but licenses were detected in source scan.

```
/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements.  See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License.  You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
    * Checks if this content model has had its min/maxOccurs values reduced for
    * purposes of speeding up UPA.  If
```

so, this content model should not be used
* for any purpose other than checking unique particle attribution
*
* @return a boolean that says whether this content has been compacted for UPA
*/

Found in path(s):
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/models/XSCMValidator.java
No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements.  See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License.  You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/YearDV.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/MonthDayDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XMLSchemaValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSModelGroupImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSAttributeChecker.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/AbstractDateTimeDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/DurationDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/TimeDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDKeyrefTraverser.java

*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/DateTimeDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/DateDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/MonthDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/XSSimpleTypeDecl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/DayDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/YearMonthDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDUniqueOrKeyTraverser.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2001 World Wide Web Consortium,
* (Massachusetts Institute of Technology, Institut National de
* Recherche en Informatique et en Automatique, Keio University). All
* Rights Reserved. This program is distributed under the W3C's Software
* Intellectual Property License. This program is distributed in the
* hope that it will be useful, but WITHOUT ANY WARRANTY; without even
* the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
* PURPOSE.
* See W3C License http://www.w3.org/Consortium/Legal/ for more details.
*/

Found in path(s):
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/ElementEditAS.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/ASElementDeclaration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/ASDataType.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/CharacterDataEditAS.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/ASModel.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/DocumentAS.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/ASContentModel.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/xerces/dom3/as/ASObjectList.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/DOMASBuilder.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/DOMASWriter.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/ASEntityDeclaration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/ASNamedObjectMap.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/ASObject.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/ASAttributeDeclaration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/DOMASException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/DOMImplementationAS.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/ASNotationDeclaration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/DocumentEditAS.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom3/as/NodeEditAS.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2000 World Wide Web Consortium,
* (Massachusetts Institute of Technology, Institut National de
* Recherche en Informatique et en Automatique, Keio University). All
* Rights Reserved. This program is distributed under the W3C's Software
* Intellectual Property License. This program is distributed in the
* hope that it will be useful, but WITHOUT ANY WARRANTY; without even
* the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
* PURPOSE. See W3C License http://www.w3.org/Consortium/Legal/ for more
* details.
*/
```

Found in path(s):
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/w3c/dom/html/HTMLDOMImplementation.java
No license file was found, but licenses were detected in source scan.

```
/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements.  See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
```

* (the "License"); you may not use this file except in compliance with
* the License.  You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
 * used to check the 3 constraints against each complex type
 * (should be each model group):
 * Unique Particle
 Attribution, Particle Derivation (Restriction),
 * Element Declrations Consistent.
 */

Found in path(s):
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSConstraints.java
No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements.  See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License.  You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
// Unique Particle Attribution

Found in path(s):
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/models/XSDFACM.java
No license file was found, but licenses were detected in source scan.

# The ASF licenses this file to You under the Apache License, Version 2.0

# (the "License"); you may not use this file except in compliance with
# the License.  You may obtain a copy of the License at
#  http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/msg/DOMMessages.properties
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/msg/XMLMessages.properties
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/msg/XMLSchemaMessages.properties
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/msg/XMLSerializerMessages.properties
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/msg/XPointerMessages.properties
*
 /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/msg/XIncludeMessages.properties
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/regex/message.properties
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/regex/message_ja.properties
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/msg/JAXPValidationMessages.properties
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/msg/SAXMessages.properties
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/msg/DatatypeMessages.properties
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/regex/message_fr.properties
No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements.  See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License.  You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

```
*/

Found in path(s):
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-
jar/org/apache/xerces/dom/DOMImplementationListImpl.java
No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements.  See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License.  You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
  * DOM Level 3 LS CR - Experimental.
   * Create a new <code>LSParser</code>. The newly constructed parser may
   *
 then be configured by means of its <code>DOMConfiguration</code>
   * object, and used to parse documents by means of its <code>parse</code>
   *  method.
   * @param mode  The <code>mode</code> argument is either
   *   <code>MODE_SYNCHRONOUS</code> or <code>MODE_ASYNCHRONOUS</code>, if
   *   <code>mode</code> is <code>MODE_SYNCHRONOUS</code> then the
   *   <code>LSParser</code> that is created will operate in synchronous
   *   mode, if it's <code>MODE_ASYNCHRONOUS</code> then the
   *   <code>LSParser</code> that is created will operate in asynchronous
   *   mode.
   * @param schemaType  An absolute URI representing the type of the schema
   *   language used during the load of a <code>Document</code> using the
   *   newly created <code>LSParser</code>. Note that no lexical checking
   *   is done on the absolute URI. In order to create a
   *   <code>LSParser</code> for any kind of schema types (i.e. the
   *   LSParser will be free to use any schema
 found), use the value
   *   <code>null</code>.
   * <p ><b>Note:</b>    For W3C XML Schema [<a href='http://www.w3.org/TR/2001/REC-xmlschema-1-
20010502/'>XML Schema Part 1</a>]
   *   , applications must use the value
```

```
   *   <code>"http://www.w3.org/2001/XMLSchema"</code>. For XML DTD [<a
href='http://www.w3.org/TR/2000/REC-xml-20001006'>XML 1.0</a>],
   *   applications must use the value
   *   <code>"http://www.w3.org/TR/REC-xml"</code>. Other Schema languages
   *   are outside the scope of the W3C and therefore should recommend an
   *   absolute URI in order to use this method.
   * @return  The newly created <code>LSParser</code> object. This
   *   <code>LSParser</code> is either synchronous or asynchronous
   *   depending on the value of the <code>mode</code> argument.
   * <p ><b>Note:</b>    By default, the newly created <code>LSParser</code>
   *    does not contain a <code>DOMErrorHandler</code>, i.e. the value of
   *   the "<a href='http://www.w3.org/TR/2003/WD-DOM-Level-3-Core-20030609/core.html#parameter-error-
handler'>
    *   error-handler</a>" configuration parameter is <code>null</code>. However, implementations
   *   may provide a default error handler at creation time. In that case,
   *   the initial value of the <code>"error-handler"</code> configuration
   *   parameter on the new created <code>LSParser</code> contains a
   *   reference to the default error handler.
   * @exception DOMException
   *    NOT_SUPPORTED_ERR: Raised if the requested mode or schema type is
   *   not supported.
  */
```

Found in path(s):
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/CoreDOMImplementationImpl.java
No license file was found, but licenses were detected in source scan.

```
/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements.  See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License.  You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XMLElementDecl.java

*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/XPath.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLTableCaptionElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/ValidatorHelper.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLComponent.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLPullParserConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/events/EventImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/XMLSchemaValidatorComponentManager.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLSetvarElementImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XMLDTDValidatorFilter.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XMLDTDDescription.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/AbstractDOMParser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSModelGroup.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/XMLSchemaFactory.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/DOMDocumentHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/ErrorHandlerWrapper.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/DayTimeDurationDV.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xinclude/ObjectFactory.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLOListElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/SecuritySupport.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLTrElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/StAXLocationWrapper.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDGroupTraverser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/xerces/jaxp/DocumentBuilderFactoryImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/StAXSchemaParser.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/StAXValidatorHelper.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/util/HexBin.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/dtd/XML11IDREFDatatypeValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/CachingParserPool.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLDListElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSWildcard.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLTimerElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDocumentInfo.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/AttributeMap.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLOneventElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/dtd/XML11DTDDVFactoryImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLPElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/EncodingMap.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLTrElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLStyleElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/XMLDTDContentModelHandler.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLDocumentSource.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/datatypes/XSDecimal.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLInputElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSElementDeclHelper.java

* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSNotationDeclaration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/ElementDefinitionImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/models/CMUniOp.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/io/UTF16Reader.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/Serializer.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/dtd/StringDatatypeValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/models/CMNodeFactory.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/SAXParser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLIElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/URI.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/ObjectFactory.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/DefaultDocument.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/SoftReferenceGrammarPool.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xpointer/ShortHandPointer.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/DefaultXMLDocumentHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/DefaultErrorHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/XMLGrammarParser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/Printer.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/XMLString.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/ValidationContext.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/HexBinaryDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLAnchorElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLDocumentImpl.java

* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDAttributeTraverser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLDTDScanner.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/JAXPValidatorComponent.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/BaseSchemaDVFactory.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/dtd/ENTITYDatatypeValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredCommentImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSAttributeDecl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/SecuritySupport.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/ListDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLTdElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLAreaElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLTemplateElementImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/EndElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XML11NSDTDValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLTableElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/dtd/XML11IDDatatypeValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/LocatorProxy.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/models/XSCMLeaf.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/DefaultNode.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/SecureProcessingConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLRefreshElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/xerces/impl/xpath/regex/RegexParser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/SecurityManager.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLGoElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/ExtendedSchemaDVFactoryImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/ItemPSVI.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DOMMessageFormatter.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/models/XSAllCM.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/CDATASectionImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLIFrameElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/util/ByteListImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/identity/KeyRef.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xinclude/XIncludeHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/validation/ValidationManager.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/util/XS10TypeHelper.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLTextAreaElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLFrameSetElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xinclude/MultipleScopeNamespaceSupport.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLDocumentScanner.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/AbstractXMLDocumentParser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredAttrNSImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSObject.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/grammars/XMLSchemaDescription.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/XMLErrorCode.java

* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XMLDTDScannerImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLStrongElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/io/ASCIIReader.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSParticle.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/regex/ParseException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/StAXEventResultBuilder.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLHeadingElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/ProcessingInstructionImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/CharactersImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLLinkElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/HTTPInputSource.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/PSVIProvider.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLTitleElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLIElement.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/XMLGrammarCachingConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLDTDSource.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredElementNSImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/EntityResolverWrapper.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSMultiValueFacet.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSAnnotationImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/datatypes/XSDateTime.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLSelectElementImpl.java
*

/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLCardElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/ObjectFactory.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLPElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLMetaElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLTdElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/grammars/XMLGrammarLoader.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLAccessElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLAnchorElementImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSAnnotationInfo.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSAttributeUseImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/SAXInputSource.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/SchemaDateTimeException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDAbstractIDConstraintTraverser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/AttrImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLIsIndexElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/NamespaceSupport.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLLabelElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xinclude/XIncludeNamespaceSupport.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSComplexTypeDefinition.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLQuoteElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/identity/IdentityConstraint.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLBaseFontElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/XMLAttributesImpl.java

* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLTableColElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLAnchorElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSImplementation.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DOMInputImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSAttributeGroupDefinition.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/XMLResourceIdentifierImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSModel.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/models/DFAContentModel.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/ProcessingInstructionImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/ValidatedInfo.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XMLEntityScanner.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/ImmutableLocation.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/EmptyLocation.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/SAXLocatorWrapper.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/PrecisionDecimalDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLInputSource.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/ValidatorHandlerImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/SAXMessageFormatter.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/models/XSEmptyCM.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XMLNSDocumentScannerImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/datatypes/XSDouble.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLOptgroupElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/xerces/jaxp/datatype/DurationImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDComplexTypeTraverser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLSelectElementImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredDocumentTypeImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/XMLDTDHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLOptionElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DocumentImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/ParserConfigurationSettings.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/grammars/XSGrammar.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XML11EntityScanner.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/DOMASBuilderImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/grammars/XMLDTDDescription.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLMenuElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/identity/Field.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/EntityReferenceImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/identity/ValueStore.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/AttrNSImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/CoreDocumentImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/ObjectFactory.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLTableRowElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xpointer/ElementSchemePointer.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/NamespaceContext.java

* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/DOMParser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xinclude/XIncludeMessageFormatter.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/StringDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLBuilder.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/SchemaContentHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLOneventElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSLoaderImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLBrElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSAnnotation.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xinclude/XIncludeTextReader.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/Method.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/XMLDocumentFragmentHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLCollectionImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/AnyAtomicDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XMLEntityHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XMLVersionDetector.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/dtd/DTDDVFactoryImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLPrevElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/Augmentations.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSConstants.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLImageElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/xerces/dom/PSVIAttrNSImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DOMOutputImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/io/UTF8Reader.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/PSVIDOMImplementationImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/util/XSGrammarPool.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLFieldsetElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/util/XInt.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLParagraphElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLBRElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/XMLCatalogResolver.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/XML11Char.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/DOMSerializerImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/XSGrammarPoolContainer.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/DefaultNamespaceContext.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/regex/Op.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLHeadElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/IntStack.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDAttributeGroupTraverser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XMLScanner.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/XML11Configurable.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/regex/Match.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/ValidatorImpl.java

* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/UnionDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSDeclarationPool.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/XMLEventImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLHtmlElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLFormControl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xpointer/XPointerHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/XSSimpleTypeDelegate.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/DTDGrammarBucket.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredEntityImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/AugmentationsImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XMLNotationDecl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLTableElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/DOMParserImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/InvalidDatatypeValueException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/models/MixedContentModel.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/DTDDVFactory.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/grammars/XMLGrammarDescription.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DOMLocatorImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSNotationDecl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/models/XSCMBinOp.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/QName.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/datatypes/XSQName.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/SchemaGrammar.java

*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/models/XSCMUniOp.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/ErrorHandlerProxy.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XMLDTDProcessor.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLMetaElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLTableSectionElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLButtonElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XMLDTDLoader.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/util/XSObjectListImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSElementDecl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLFormElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/AttributePSVImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLRefreshElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/SchemaParsingConfig.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/regex/CaseInsensitiveMap.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/models/CMAny.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/ElementImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/ElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XMLDocumentScannerImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/XMLLocatorWrapper.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/IndentPrinter.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDElementTraverser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/WeakReferenceXMLSchema.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/xerces/parsers/SoftReferenceSymbolTableConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/DOMEntityResolverWrapper.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSLoader.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/PSVIErrorList.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/regex/ParserForXMLSchema.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/JAXPValidationMessageFormatter.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/IntegratedParserConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLEmElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSValue.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/AnySimpleDV.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/PSVIElementNSImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XMLErrorReporter.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/ParentNode.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLDTDFilter.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/DOMValidatorHelper.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/NodeImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSMessageFormatter.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/XML11NonValidatingConfiguration.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/NamedNodeMapImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredAttrImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/TextSerializer.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSComplexTypeDecl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/DefaultText.java

* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XMLSchemaException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/DOMUtil.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSParticleDecl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/XMLLocator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/SynchronizedSymbolTable.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/XPointerParserConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLBElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/JAXPConstants.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/ASDOMImplementationImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLNoopElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XMLNSDTDValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredDocumentImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XMLSchemaLoader.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/BasicParserConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/SAXParserImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredProcessingInstructionImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLAppletElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSImplementationImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/TreeWalkerImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/DOMErrorHandlerWrapper.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DOMImplementationImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/BalancedDTDGrammar.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/xerces/dom/NodeIteratorImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLDocument.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XMLEntityDescription.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/datatype/SerializedXMLGregorianCalendar.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DOMStringListImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/ChildNode.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/YearMonthDurationDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/ShortList.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLHeadElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLInputElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/XML11Configuration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/SerializerFactoryImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XML11NSDocumentScannerImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/dtd/NMTOKENDatatypeValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/StAXInputSource.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/IDREFDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/XMLGrammarPoolImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/ExternalSubsetResolver.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/XHTMLSerializer.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLTableElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/DefaultValidationErrorHandler.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/DatatypeValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/NamedNodeMapImpl.java

* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/regex/BMPattern.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/util/XSNamedMapImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredTextImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/XMLGrammarPreparser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/datatype/SerializedDuration.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/SimpleXMLSchema.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLConfigurationException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DOMErrorImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/XPathException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLEmElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DOMImplementationSourceImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/ElementNSImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSAttributeUse.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/RangeExceptionImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDAbstractParticleTraverser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/dtd/NOTATIONDatatypeValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/XMLSchema.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/regex/RangeToken.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLParamElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLScriptElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/util/XIntPool.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLAElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/xerces/impl/xs/FilePathToURI.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/SchemaDOMParser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLInputElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLLegendElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLOptionElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLHeadElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLFieldsetElementImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLPostfieldElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/models/CMBinOp.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/SchemaNamespaceSupport.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/XSSimpleType.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLTimerElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLBElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/AttributeImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLMapElementImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/DTDImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DOMConfigurationImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredElementDefinitionImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLLIElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/XMLChar.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLObjectElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLNoopElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/ASModelImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/xml/serialize/Encodings.java
*
 /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XML11DTDValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/TypeValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLTableCellElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLBaseElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLPrevElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLDOMImplementationImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredDOMImplementationImpl.java
*
 /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/StartElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XMLNamespaceBinder.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/io/Latin1Reader.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/events/MutationEventImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLDOMImplementation.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/ShadowedSymbolTable.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLAccessElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/AttributePSVI.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLMetaElement.java
*
 /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLImgElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/validation/ConfigurableValidationState.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/SchemaDVFactoryImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredCDATASectionImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLSmallElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDSimpleTypeTraverser.java

* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/Constants.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLSelectElement.java
*
 /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/XML11Serializer.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/XMLSerializer.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/msg/XMLMessageFormatter.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/UnparsedEntityHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/QNameDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/SerializerFactory.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/DOMSerializer.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/XMLParser.java
*
 /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xinclude/SecuritySupport.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/DatatypeMessageFormatter.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSModelImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLDocumentImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/ObjectFactory.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredNotationImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xinclude/XInclude11TextReader.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/LCount.java
*
 /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/StAXStreamResultBuilder.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XML11NamespaceBinder.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/datatypes/XSFloat.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/models/ContentModelValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/xerces/impl/dv/xs/DoubleDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/regex/RegularExpression.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLUElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/EntityResolver2Wrapper.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/models/XSCMRepeatingLeaf.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLWmlElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSElementDeclaration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/util/LSInputListImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/XNIException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/validation/ValidationState.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLWmlElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/EntityImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/RevalidationHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/CharacterDataImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/TeeXMLDocumentFilterImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/SAXParserFactoryImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/EndDocumentImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/PSVIDocumentImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/XMLDocumentHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/Base64BinaryDV.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLHRElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/ElementState.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/CommentImpl.java

* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/models/CMLeaf.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/SchemaSymbols.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/IntegerDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XMLContentSpec.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/NodeImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLPostfieldElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/HTMLSerializer.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/DTDConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/Version.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/SymbolTable.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLPreElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/io/UCSReader.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/ReadOnlyGrammarPool.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/FullDVFactory.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/DOMResultAugmentor.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/SchemaDOM.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/SchemaDVFactory.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/AttributesProxy.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSFacet.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XMLEntityManager.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/util/StringListImpl.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSSimpleTypeDefinition.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/xerces/parsers/XMLDocumentParser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/SoftReferenceSymbolTable.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/util/XSNamedMap4Types.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/identity/FieldActivator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSGrammarBucket.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLParserConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/datatypes/ObjectList.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/regex/Token.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/NamespaceImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSAttributeDeclaration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/models/CMNode.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XMLDocumentFragmentScannerImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLStrongElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/DOMInputSource.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/SecuritySupport.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/XSFacets.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xpointer/XPointerProcessor.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLImgElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XMLAttributeDecl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/DTDGrammar.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLSmallElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDNotationTraverser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLDocumentFilter.java
*

/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/NodeListCache.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredNode.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/ElementPSVImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/MessageFormatter.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSTypeDefinition.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLBigElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/EntityDeclarationImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xpointer/XPointerPart.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/CommentImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/util/SimpleLocator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/models/SimpleContentModel.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLDOMImplementationImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLFieldSetElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XMLDTDValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DocumentTypeImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/dtd/ListDatatypeValidator.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/SecuritySupport.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/ElementPSVI.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLAElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/BaseMarkupSerializer.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/XMLStringBuffer.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSAttributeGroupDecl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DocumentFragmentImpl.java

* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSGroupDecl.java
*
 /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DOMNormalizer.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLUListElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xpointer/XPointerErrorHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLOptionElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/NotationImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/AbstractSAXParser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/XIncludeParserConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XMLSimpleType.java
*
 /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/PrimeNumberSequenceGenerator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDAbstractTraverser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/NotationDeclarationImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XML11DTDProcessor.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLDTDContentModelSource.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/DecimalDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/SecuritySupport.java
*
 /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/LineSeparator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/SubstitutionGroupHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLOptGroupElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/AbstractXMLSchema.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLSetvarElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLFontElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/xerces/impl/validation/EntityState.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLCardElement.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLComponentManager.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSDDescription.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeferredEntityReferenceImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/DOMResultBuilder.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/JAXPNamespaceContextWrapper.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/util/XSInputSource.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/EmptyXMLSchema.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSIDCDefinition.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/XML11DTDConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/StandardParserConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/AttrImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/datatype/XMLGregorianCalendarImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/SchemaDOMImplementation.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/DTDParser.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLBodyElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/dtd/XML11NMTOKENDatatypeValidator.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/StAXDocumentHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/InvalidDatatypeFacetException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/ObjectFactory.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/grammars/XMLGrammarPool.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/StringList.java

* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/identity/UniqueOrKey.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/XMLSymbols.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLOptgroupElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLErrorHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/identity/Selector.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/EncodingInfo.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/BaseDVFactory.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/dtd/IDDatatypeValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XML11DocumentScannerImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/DVFactoryException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLDTDContentModelFilter.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/StreamValidatorHelper.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/LSInputList.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/models/CMBuilder.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/FloatDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/datatypes/ByteList.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/BooleanDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/ElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/events/UIEventImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/grammars/Grammar.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLBigElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/identity/XPathMatcher.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/xerces/parsers/NonValidatingConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xpath/regex/REUtil.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/XSWildcardDecl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSNamespaceItem.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLDivElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/AnyURIDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/DatatypeException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSModelGroupDefinition.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/NameNodeListImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DOMXSImplementationSourceImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/events/MouseEventImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSObjectList.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/TextImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLGoElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLParseException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/SchemaValidatorConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/SymbolHash.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLModElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/util/ObjectListImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLBrElement.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSTerm.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/parser/XMLEntityResolver.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/models/CMStateSet.java

* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/io/MalformedByteSequenceException.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/XMLEventFactoryImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/datatype/DatatypeFactoryImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/SecurityConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLDoElement.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/RangeImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dtd/XMLEntityDecl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/OutputFormat.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/util/XMLEntityDescriptionImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLTemplateElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSNamedMap.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/DeepNodeListImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/EntityReferenceImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xpointer/XPointerMessageFormatter.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/EntityDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/parsers/XIncludeAwareParserConfiguration.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/WMLUElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/XML11DTDScannerImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/DraconianErrorHandler.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/stax/events/StartDocumentImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/traversers/XSDWildcardTraverser.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/util/ShortListImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-

jar/org/apache/html/dom/HTMLFrameElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/dtd/IDREFDatatypeValidator.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/html/dom/HTMLDirectoryElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/xs/opti/DefaultElement.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/DocumentBuilderImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xml/serialize/HTMLdtd.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xs/XSNamespaceItemList.java
*
/opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/util/Base64.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/dom/TextImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/XMLAttributes.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/jaxp/validation/Util.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/wml/dom/WMLDoElementImpl.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/impl/dv/xs/IDDV.java
* /opt/cola/permits/1110637609_1606872750.0/0/xercesimpl-2-12-0-sources-2-jar/org/apache/xerces/xni/XMLResourceIdentifier.java

# 1.62 guava 29.0-jre

## 1.62.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Equivalence.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ContiguousSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/net/package-info.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ThreadFactoryBuilder.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Strings.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Ascii.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/UncaughtExceptionHandlers.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/Monitor.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ListeningExecutorService.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/SortedLists.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ForwardingBlockingQueue.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/Atomics.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/annotations/package-info.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/package-info.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/annotations/Beta.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingDeque.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/TransformedIterator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ForwardingBlockingDeque.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/FilteredKeySetMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractSortedKeySortedSetMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingNavigableSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/TransformedListIterator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingBlockingDeque.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingImmutableList.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/CompactHashSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/FilteredKeyListMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/RangeMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/SortedMultisetBridge.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/FilteredMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/UnmodifiableSortedMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/TreeRangeMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/RegularImmutableAsList.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/DescendingImmutableSortedSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-

jar/com/google/common/collect/FilteredEntrySetMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/ForwardingImmutableMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/AllEqualOrdering.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/ForwardingImmutableSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/CompactLinkedHashSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/AbstractNavigableMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/FilteredSetMultimap.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/CompactLinkedHashMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/EvictingQueue.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/FilteredEntryMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/ImmutableEnumMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/ForwardingNavigableMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/DescendingMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/CompactHashMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/TreeTraverser.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/Reflection.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for extra methods of {@code Objects} only in web. Intended to be empty for regular
* version.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/ExtraObjectsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Joiner.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/MultiReader.java

* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ListenableFutureTask.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/FluentIterable.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/FileBackedOutputStream.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/Bytes.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/Ints.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Stopwatch.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/Chars.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/Shorts.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/Longs.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/escape/UnicodeEscaper.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/escape/Escaper.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/net/InetAddresses.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/net/PercentEscaper.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/Booleans.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/internal/Finalizer.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Converter.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/thirdparty/publicsuffix/TrieParser.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/Doubles.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/SequentialExecutor.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/CharMatcher.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/Floats.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Ints}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/primitives/IntsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/BaseImmutableMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/JdkBackedImmutableMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/JdkBackedImmutableBiMap.java
*

/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/IndexedImmutableSet.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Shorts}. Intended to be empty for regular
* version.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/ShortsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/DirectExecutor.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Interners.java

* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/MultiInputStream.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/eventbus/Subscribe.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/EnumMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Function.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/eventbus/AsyncEventBus.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Defaults.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/Files.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/eventbus/EventBus.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/CountingInputStream.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/Flushables.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/FinalizableReference.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/package-info.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Predicate.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/CountingOutputStream.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/FinalizableSoftReference.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Throwables.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ListenableFuture.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/package-info.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/eventbus/package-info.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/package-info.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/FinalizableWeakReference.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/ByteStreams.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/eventbus/AllowConcurrentEvents.java

* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/LineBuffer.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/LittleEndianDataOutputStream.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Preconditions.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/CharStreams.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Functions.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/FinalizablePhantomReference.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/LittleEndianDataInputStream.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Charsets.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/Closeables.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/Primitives.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Suppliers.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/Resources.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Objects.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Supplier.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/AbstractFuture.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/AbstractIterator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/HashBiMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/FinalizableReferenceQueue.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/LineReader.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/eventbus/DeadEvent.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Predicates.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ExecutionList.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/CompactHashing.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/IncidentEdgeSet.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/StandardTable.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/HashBasedTable.java

*
/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Collections2.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableListMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/RegularImmutableBiMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableMapEntrySet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/UnmodifiableIterator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableCollection.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Serialization.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/EmptyImmutableListMultimap.java
*
/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/PeekingIterator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableMapValues.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Table.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/thirdparty/publicsuffix/PublicSuffixPatterns.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableEntry.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableMapKeySet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableSortedSet.java
*
/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/CollectPreconditions.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableBiMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Tables.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Range.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/StandardRowSortedTable.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-

jar/com/google/common/collect/SingletonImmutableBiMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/TreeBasedTable.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/Platform.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/RegularImmutableMap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/base/PatternCompiler.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/base/CommonPattern.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/base/JdkPattern.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/base/CommonMatcher.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Outer class that exists solely to let us write {@code Partially.GwtIncompatible} instead of plain
* {@code GwtIncompatible}. This is more accurate for {@link Futures#catching}, which is available
* under GWT but with a slightly different signature.
*
* <p>We can't use {@code PartiallyGwtIncompatible} because then the GWT compiler
 wouldn't recognize
* it as a {@code GwtIncompatible} annotation. And for {@code Futures.catching}, we need the GWT
* compiler to autostrip the normal server method in order to expose the special, inherited GWT
* version.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/util/concurrent/Partially.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/CollectSpliterators.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/ImmutableBiMapFauxverideShim.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/graph/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Doubles}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/DoublesMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/ReaderInputStream.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/CombinedFuture.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/AggregateFutureState.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/AsyncCallable.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ConsumingQueueIterator.java

* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/MacHashFunction.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/InterruptibleTask.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/Platform.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/FarmHashFingerprint64.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/LittleEndianByteArray.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not
* use this file except in compliance with the License. You may obtain a copy of
* the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/SortedMultisets.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/SortedMultiset.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.

```
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/MoreObjects.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/TrustedListenableFutureTask.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/eventbus/Dispatcher.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ListenerCallQueue.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/math/Quantiles.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/eventbus/Subscriber.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/eventbus/SubscriberRegistry.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
  * This method was written by Doug Lea with assistance from members of JCP JSR-166 Expert Group
  * and released to the public domain, as explained at
  * http://creativecommons.org/licenses/publicdomain
  *
  * As of 2010/06/11, this method is identical to the (package private) hash method in OpenJDK 7's
  * java.util.HashMap
 class.
  */
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/Striped.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 The Guava Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/package-info.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ExecutionSequencer.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/ImmutableSupplier.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/JdkBackedImmutableSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/JdkBackedImmutableMultiset.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/ImmutableLongArray.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/AbstractHashFunction.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ForwardingCondition.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/ImmutableIntArray.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/ImmutableDoubleArray.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ForwardingLock.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/Platform.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/Internal.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/GwtTransient.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/AtomicLongMap.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/SortedIterable.java

* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/RangeSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/SortedIterables.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/GeneralRange.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/RegularImmutableSortedMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableSortedMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingSortedMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Count.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableSortedMultisetFauxverideShim.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractRangeSet.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you
* may not use this file except in compliance with the License.  You may
* obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
* implied.  See the License for the specific language governing
* permissions and limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Streams.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
```

```
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
  * Returns an array containing all of the elements in the specified collection. This method
  * returns the elements in the order they are returned by the collection's iterator. The returned
  * array is "safe" in that no references to it are maintained by the collection. The caller is
  * thus free to modify the returned
 array.
  *
  * <p>This method assumes that the collection size doesn't change while the method is running.
  *
  * <p>TODO(kevinb): support concurrently modified collections?
  *
  * @param c the collection for which to return an array of elements
  */
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ObjectArrays.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/ImmutableGraph.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-

jar/com/google/common/graph/SuccessorsFunction.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/graph/Network.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/io/RecursiveDeleteOption.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/graph/MutableNetwork.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/graph/PredecessorsFunction.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/TopKSelector.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/graph/ImmutableNetwork.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/graph/Graph.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/graph/MutableGraph.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/io/InsecureRecursiveDeleteException.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/graph/Graphs.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/io/CharSequenceReader.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/util/concurrent/Runnables.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/FilteredMultimapValues.java
*

/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/util/concurrent/WrappingScheduledExecutorService.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/base/Utf8.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/hash/HashingInputStream.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/AbstractTable.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/reflect/TypeVisitor.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/base/VerifyException.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/eventbus/SubscriberExceptionHandler.java
*
/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/base/Verify.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/eventbus/SubscriberExceptionContext.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Floats}. Intended to be empty for regular
* version.
*/
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/primitives/FloatsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
  * This method was rewritten in Java from an intermediate step of the Murmur hash function in
  * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
  * following header:
  *
  * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
  * hereby disclaims
 copyright to this source code.
  */

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/base/SmallCharMatcher.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/ImmutableMapEntry.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-

jar/com/google/common/io/MoreFiles.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/MultimapBuilder.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
  * This following method is a modified version of one found in
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30
  * which contained the following notice:
  *
  * Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to
  *
 the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/
  *
  * Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.
  */
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/util/concurrent/MoreExecutors.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/StandardMutableValueGraph.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/AbstractGraph.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/StandardNetwork.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/MutableValueGraph.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/MapRetrievalCache.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableMultisetGwtSerializationDependencies.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/AbstractGraphBuilder.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/EndpointPairIterator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/CollectCollectors.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/GraphBuilder.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/GraphConnections.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/DirectedGraphConnections.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/EndpointPair.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/ForwardingGraph.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/NetworkConnections.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/ForwardingNetwork.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/MultiEdgesConnecting.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/AbstractUndirectedNetworkConnections.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/RangeGwtSerializationDependencies.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/ForwardingValueGraph.java

* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/AbstractValueGraph.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/MapIteratorCache.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ArrayListMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/ValueGraphBuilder.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/AbstractNetwork.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/UndirectedMultiNetworkConnections.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/ValueGraph.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/NetworkBuilder.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/UndirectedNetworkConnections.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/DirectedNetworkConnections.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/StandardMutableNetwork.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/LinkedHashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/StandardValueGraph.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/DirectedMultiNetworkConnections.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/HashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/EdgesConnecting.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/StandardMutableGraph.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/AbstractDirectedNetworkConnections.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/UndirectedGraphConnections.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/GraphConstants.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/MoreCollectors.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Comparators.java
*

/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/ImmutableValueGraph.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/ElementOrder.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/MinMaxPriorityQueue.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/SortedMapDifference.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/RowSortedTable.java
*
/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingImmutableCollection.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingSortedSetMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractSequentialIterator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/UnmodifiableListIterator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingSetMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingListMultimap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/RateLimiter.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/LongAddables.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/math/StatsAccumulator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/TypeToInstanceMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/Element.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/ByteSink.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/FileWriteMode.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/escape/package-info.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/math/PairedStatsAccumulator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/SipHashFunction.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/AbstractByteHasher.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/CharSink.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/html/package-info.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/ChecksumHashFunction.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/LongAddable.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableRangeSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/package-info.java

* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/math/LinearTransformation.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/MutableTypeToInstanceMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/CartesianList.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/ByteSource.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/ImmutableTypeToInstanceMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/FilteredKeyMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/LongAddable.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableRangeMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/BaseEncoding.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/AbstractInvocationHandler.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/CharSource.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/ClassPath.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/StandardSystemProperty.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/Closer.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/LongAddables.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ServiceManager.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/SmoothRateLimiter.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ListenableScheduledFuture.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/math/PairedStats.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/TypeCapture.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/xml/package-info.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/Parameter.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/Invokable.java

* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/math/Stats.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
  * Not supported. <b>You are attempting to create a map that may contain a non-{@code Comparable}
  * key.</b> Proper calls will resolve to the version in {@code ImmutableSortedMap}, not this dummy
  * version.
  *
  * @throws UnsupportedOperationException always
  * @deprecated <b>Pass a key of type {@code Comparable}
 to use {@link
  *     ImmutableSortedMap#of(Comparable, Object)}.</b>
  */
```

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableSortedMapFauxverideShim.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
```

*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/RemovalListener.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractSortedMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/HashingOutputStream.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/math/IntMath.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/UnsignedLongs.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/FunctionalEquivalence.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/math/LongMath.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/PrimitiveSink.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/HashFunction.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Present.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/EmptyContiguousSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/CacheStats.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/Hasher.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Enums.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/LoadingCache.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/Murmur3_128HashFunction.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/BloomFilterStrategies.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/math/MathPreconditions.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/RemovalNotification.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/RemovalCause.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ExecutionError.java

* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/net/MediaType.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/AsyncFunction.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/net/HttpHeaders.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Ticker.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/PairwiseEquivalence.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ForwardingListeningExecutorService.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/AbstractLoadingCache.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/ForwardingLoadingCache.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/math/DoubleUtils.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Absent.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ForwardingExecutorService.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/TypeParameter.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/math/DoubleMath.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/UnsignedInteger.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/Types.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/ForwardingCache.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/Funnels.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/Funnel.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/MessageDigestHashFunction.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/package-info.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/RegularContiguousSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/AbstractListeningExecutorService.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/UnsignedInts.java

*
/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/hash/Crc32cHashFunction.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/cache/CacheLoader.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/hash/AbstractHasher.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/util/concurrent/ListeningScheduledExecutorService.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/hash/Murmur3_32HashFunction.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/TreeRangeSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/RegularImmutableMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/hash/AbstractNonStreamingHashFunction.java
*
/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/math/package-
info.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/cache/Weigher.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/util/concurrent/CycleDetectingLockFactory.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/util/concurrent/AbstractScheduledService.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/cache/AbstractCache.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/DescendingImmutableSortedMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/primitives/ParseRequest.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/util/concurrent/UncheckedExecutionException.java
*
/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/util/concurrent/Uninterruptibles.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/net/HostAndPort.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/hash/AbstractCompositeHashFunction.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/base/Optional.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/Queues.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/hash/AbstractStreamingHasher.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-

jar/com/google/common/util/concurrent/WrappingExecutorService.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/hash/BloomFilter.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/hash/Hashing.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/math/BigIntegerMath.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/cache/CacheBuilderSpec.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/BoundType.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/primitives/UnsignedLong.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/util/concurrent/FutureCallback.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/cache/RemovalListeners.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/hash/HashCode.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/cache/Cache.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/*
  * This method was rewritten in Java from an intermediate step of the Murmur hash function in
  * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
  * following header:
  *
  * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
  * hereby
```

disclaims copyright to this source code.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Hashing.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/IgnoreJRERequirement.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/FluentFuture.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/AbstractCatchingFuture.java

* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/AggregateFuture.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/escape/CharEscaper.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ImmediateFuture.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/CollectionFuture.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/PatternFilenameFilter.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/CaseFormat.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/annotations/VisibleForTesting.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/TimeoutFuture.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/GwtFluentFutureCatchingSpecialization.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/TimeLimiter.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/FuturesGetChecked.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/AppendableWriter.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/SimpleTimeLimiter.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/FakeTimeLimiter.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/TypeToken.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/escape/CharEscaperBuilder.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/AbstractTransformFuture.java
No license file was found, but licenses were detected in source scan.

/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at

* http://creativecommons.org/publicdomain/zero/1.0/
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/LongAdder.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/hash/Striped64.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/Striped64.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/LongAdder.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/AtomicDoubleArray.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/RegularImmutableSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ExplicitOrdering.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractSetMultimap.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractMapBasedMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingSortedMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/HashMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ClassToInstanceMap.java

* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Iterators.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/LinkedHashMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingConcurrentMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingObject.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/TreeMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/MapDifference.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/UsingToStringOrdering.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/SetMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractSortedSetMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingCollection.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Sets.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/TreeMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ConcurrentHashMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/CompoundOrdering.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Ordering.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ReverseOrdering.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/SingletonImmutableSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/EnumHashBiMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ByFunctionOrdering.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingList.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/HashMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-

jar/com/google/common/collect/NullsFirstOrdering.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingMapEntry.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/LinkedHashMultiset.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Multimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Multisets.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingSortedSet.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/MutableClassToInstanceMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Interner.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/BiMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/EnumBiMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingIterator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/package-info.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableList.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Synchronized.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Lists.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractMapEntry.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ComparatorOrdering.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ListMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingListIterator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractBiMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ReverseNaturalOrdering.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractListMultimap.java
*

/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/LinkedListMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Multimaps.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/LexicographicalOrdering.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractIterator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ArrayListMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Iterables.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingMultiset.java
*
/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/SortedSetMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Maps.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/NaturalOrdering.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/NullsLastOrdering.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingQueue.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractMapBasedMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingMultimap.java
*
/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Multiset.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/BaseGraph.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/Traverser.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/graph/AbstractBaseGraph.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ArrayTable.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ForwardingTable.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableSortedMap.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/DiscreteDomain.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ComputationException.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/AbstractIndexedListIterator.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/ImmutableClassToInstanceMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-

jar/com/google/common/collect/ImmutableAsList.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/SingletonImmutableTable.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/EmptyImmutableSetMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/ImmutableTable.java
*
/opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/RegularImmutableList.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/SingletonImmutableList.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/ImmutableSetMultimap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/ImmutableEnumSet.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/ImmutableSortedSetFauxverideShim.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/ComparisonChain.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/RegularImmutableSortedSet.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-
jar/com/google/common/collect/ImmutableSortedAsList.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ServiceManagerBridge.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/LineProcessor.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/CacheBuilder.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/reflect/TypeResolver.java
*
 /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/MapMaker.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ForwardingFuture.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/net/UrlEscapers.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/net/InternetDomainName.java

* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/html/HtmlEscapers.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/ReferenceEntry.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Platform.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ForwardingFluentFuture.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/Callables.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/annotations/GwtIncompatible.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/ByteProcessor.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/escape/ArrayBasedEscaperMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/escape/Platform.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/UnsignedBytes.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/ByteArrayDataInput.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/xml/XmlEscapers.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/AbstractExecutionThreadService.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/escape/ArrayBasedCharEscaper.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/primitives/SignedBytes.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/Cut.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/MapMakerInternalMap.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/SettableFuture.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/AbstractService.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/base/Splitter.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/Service.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/annotations/GwtCompatible.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-

jar/com/google/common/escape/ArrayBasedUnicodeEscaper.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/net/HostSpecifier.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/cache/LocalCache.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/DenseImmutableTable.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/escape/Escapers.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/JdkFutureAdapters.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/io/ByteArrayDataOutput.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/ForwardingListenableFuture.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/util/concurrent/AbstractIdleService.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/SparseImmutableTable.java
* /opt/cola/permits/1110737159_1606846951.59/0/guava-29-0-jre-sources-3-jar/com/google/common/collect/RegularImmutableTable.java

# 1.63 jsoup 1.13.1

## 1.63.1 Available under license :

The MIT License

Copyright (c) 2009-2020 Jonathan Hedley <https://jsoup.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.

# 1.64 json 20180813

## 1.64.1 Available under license :

==========

Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES
OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

# 1.65 bean-validation-api 1.1.0.Final

## 1.65.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* JBoss, Home of Professional Open Source
* Copyright 2011-2013, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a
* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/ParameterDescriptor.java
No license file was found, but licenses were detected in source scan.

/*
* JBoss, Home of Professional Open Source
* Copyright 2013, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a
* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/CrossParameterDescriptor.java
*
 /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/package-info.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraintvalidation/SupportedValidationTarget.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraintvalidation/ValidationTarget.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/executable/package-info.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraintvalidation/package-info.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/bootstrap/package-info.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraints/package-info.java
*

/opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/CascadableDescriptor.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/GroupConversionDescriptor.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/package-info.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ConstraintTarget.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/spi/package-info.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2009-2012, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a
* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/Configuration.java
No license file was found, but licenses were detected in source scan.

```
/*
* JBoss, Home of Professional Open Source
* Copyright 2013, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a
* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
```

*/

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/MethodType.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012-2013, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a
* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/BootstrapConfiguration.java
No license file was found, but licenses were detected in source scan.

/*
* JBoss, Home of Professional Open Source
* Copyright 2012, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a
* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ParameterNameProvider.java

*
/opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/ConstructorDescriptor.java
No license file was found, but licenses were detected in source scan.

/*
* JBoss, Home of Professional Open Source
* Copyright 2009, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a
* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/Valid.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraints/Past.java
*
/opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/bootstrap/GenericBootstrap.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/bootstrap/ProviderSpecificBootstrap.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/Payload.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ConstraintValidator.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/ConstraintDescriptor.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraints/Null.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraints/Min.java
*
/opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraints/Pattern.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/spi/ValidationProvider.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-

jar/javax/validation/constraints/AssertFalse.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/ElementDescriptor.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/Constraint.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ConstraintDefinitionException.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraints/Future.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraints/Digits.java
*
/opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/Scope.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/UnexpectedTypeException.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraints/NotNull.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraints/AssertTrue.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ValidationProviderResolver.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/GroupDefinitionException.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ConstraintDeclarationException.java
*
/opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraints/Max.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/OverridesAttribute.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ValidationException.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/groups/Default.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/spi/BootstrapState.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/Validation.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/PropertyDescriptor.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/TraversableResolver.java
*
/opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraints/Size.java
No license file was found, but licenses were detected in source scan.

/*
* JBoss, Home of Professional Open Source
* Copyright 2012-2013, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a
* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/ExecutableDescriptor.java
No license file was found, but licenses were detected in source scan.

/*
* JBoss, Home of Professional Open Source
* Copyright 2009, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a
* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/GroupSequence.java
No license file was found, but licenses were detected in source scan.

/*
* JBoss, Home of Professional Open Source
* Copyright 2011, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a

* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/MethodDescriptor.java
No license file was found, but licenses were detected in source scan.

/*
* JBoss, Home of Professional Open Source
* Copyright 2012-2013, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a
* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ElementKind.java
*
 /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/executable/ExecutableType.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/executable/ExecutableValidator.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/ReturnValueDescriptor.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/executable/ValidateOnExecution.java

No license file was found, but licenses were detected in source scan.

```
/*
* JBoss, Home of Professional Open Source
* Copyright 2009-2013, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a
* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ConstraintViolation.java
*
 /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraints/DecimalMax.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/groups/package-info.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ConstraintValidatorContext.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/Path.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/Validator.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ReportAsSingleViolation.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/MessageInterpolator.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/metadata/BeanDescriptor.java
*
 /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/constraints/DecimalMin.java
No license file was found, but licenses were detected in source scan.

```
/*
* JBoss, Home of Professional Open Source
* Copyright 2009-2012, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a
```

* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ConstraintViolationException.java
*
 /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/spi/ConfigurationState.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ConstraintValidatorFactory.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ValidatorContext.java
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/ValidatorFactory.java
No license file was found, but licenses were detected in source scan.

/*
* JBoss, Home of Professional Open Source
* Copyright 2012, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag. See the copyright.txt in the distribution for a
* full listing of individual contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1114844475_1607559692.86/0/validation-api-1-1-0-final-sources-5-jar/javax/validation/groups/ConvertGroup.java

# 1.66 msg-simple 1.2

## 1.66.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by
the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters,
data structure
layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.
This software is dual-licensed under:

- the Lesser General Public License (LGPL) version 3.0 or, at your option, any later version;
- the Apache Software License (ASL) version 2.0.

The text of both licenses is included (under the names LGPL-3.0.txt and ASL-2.0.txt respectively).

Direct link to the sources:

- LGPL 3.0: https://www.gnu.org/licenses/lgpl-3.0.txt
- ASL 2.0: http://www.apache.org/licenses/LICENSE-2.0.txt

Apache License
Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.67 btf 1.3

## 1.67.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by
the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an

Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters,
data structure
layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the

Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

This software is dual-licensed under:

- the Lesser General Public License (LGPL) version 3.0 or, at your option, any later version;
- the Apache Software License (ASL) version 2.0.

The text of both licenses is included (under the names LGPL-3.0.txt and ASL-2.0.txt respectively).

Direct link to the sources:

- LGPL 3.0: https://www.gnu.org/licenses/lgpl-3.0.txt
- ASL 2.0: http://www.apache.org/licenses/LICENSE-2.0.txt

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.68 joda-time 2.10.9

## 1.68.1 Available under license :

```
=============================================================================
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
=============================================================================
```

This product includes software developed by
Joda.org (https://www.joda.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.69 guava 30.1-jre

## 1.69.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/SubscriberRegistry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/Dispatcher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/MoreObjects.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ListenerCallQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/math/Quantiles.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/Subscriber.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/util/concurrent/TrustedListenableFutureTask.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/AppendableWriter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/TypeToken.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/FluentFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AggregateFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/TimeLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractCatchingFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractTransformFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/FakeTimeLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/FuturesGetChecked.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/GwtFluentFutureCatchingSpecialization.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/CollectionFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/base/CaseFormat.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/escape/CharEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/PatternFilenameFilter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/escape/CharEscaperBuilder.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/TimeoutFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ImmediateFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/SimpleTimeLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/annotations/VisibleForTesting.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ServiceManagerBridge.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Java8Usage.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/reflect/TypeVisitor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/HashingInputStream.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/VerifyException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/SubscriberExceptionHandler.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/AbstractTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Runnables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Verify.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/SubscriberExceptionContext.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/FilteredMultimapValues.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/CharSequenceReader.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Utf8.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/WrappingScheduledExecutorService.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/BaseImmutableMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/IndexedImmutableSet.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
  * This following method is a modified version of one found in
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30
  * which contained the following notice:
  *
  * Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to
  *
 the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/
  *
  * Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.
  */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/MoreExecutors.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredKeySetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredKeyListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AllEqualOrdering.java
*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TransformedListIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredEntrySetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SortedMultisetBridge.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CompactLinkedHashSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CompactLinkedHashMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CompactHashMap.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractSortedKeySortedSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableEnumMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/UnmodifiableSortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingBlockingDeque.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingImmutableList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingBlockingDeque.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingDeque.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TransformedIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingImmutableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TreeTraverser.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CompactHashSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/DescendingMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredEntryMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RangeMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingNavigableSet.java

*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TreeRangeMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/DescendingImmutableSortedSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableAsList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/EvictingQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingNavigableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractNavigableMap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for extra methods of {@code Objects} only in web. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/ExtraObjectsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Tables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/thirdparty/publicsuffix/PublicSuffixPatterns.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Serialization.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/StandardTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/PeekingIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SingletonImmutableBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableMapValues.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Collections2.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableMapEntrySet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableCollection.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Table.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/EmptyImmutableListMultimap.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CollectPreconditions.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashBasedTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Platform.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TreeBasedTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/StandardRowSortedTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Range.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/UnmodifiableIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableMapKeySet.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableMapEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/collect/MultimapBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/MoreFiles.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Shorts}. Intended to be empty for regular
* version.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/ShortsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
  * This method was rewritten in Java from an intermediate step of the Murmur hash function in
  * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
  * following header:
  *
```

```
  * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
  * hereby disclaims
 copyright to this source code.
  */
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/SmallCharMatcher.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableRangeSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/LinearTransformation.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/CharSink.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/LongAddable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/SmoothRateLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredKeyMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/Parameter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ByteSink.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/Closer.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/Stats.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/html/package-info.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/PairedStatsAccumulator.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/Element.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/CharSource.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/ImmutableTypeToInstanceMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/StatsAccumulator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/MutableTypeToInstanceMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/SipHashFunction.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/TypeCapture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/LongAddable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ByteSource.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/ChecksumHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/Invokable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/RateLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/StandardSystemProperty.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/AbstractByteHasher.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/AbstractInvocationHandler.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/BaseEncoding.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/TypeToInstanceMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/PairedStats.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/LongAddables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/util/concurrent/ListenableScheduledFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/LongAddables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/CartesianList.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/FileWriteMode.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/reflect/ClassPath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/xml/package-
info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableRangeMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ServiceManager.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/AbstractBaseGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/BaseGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/Traverser.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ClosingFuture.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors

```
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Outer class that exists solely to let us write {@code Partially.GwtIncompatible} instead of plain
* {@code GwtIncompatible}. This is more accurate for {@link Futures#catching}, which is available
* under GWT but with a slightly different signature.
*
* <p>We can't use {@code PartiallyGwtIncompatible} because then the GWT compiler
 wouldn't recognize
* it as a {@code GwtIncompatible} annotation. And for {@code Futures.catching}, we need the GWT
* compiler to autostrip the normal server method in order to expose the special, inherited GWT
* version.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Partially.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/*
  * This method was rewritten in Java from an intermediate step of the Murmur hash function in
  * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
  * following header:
```

```
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby
disclaims copyright to this source code.
 */
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Hashing.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TopKSelector.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ImmutableGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/SuccessorsFunction.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/MutableNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ImmutableNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/Graphs.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/InsecureRecursiveDeleteException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/Graph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/RecursiveDeleteOption.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/graph/PredecessorsFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/MutableGraph.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/Network.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/reflect/Reflection.java
No license file was found, but licenses were detected in source scan.

/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AtomicDoubleArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/LongAdder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/LongAdder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/Striped64.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/Striped64.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/FileBackedOutputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Shorts.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Doubles.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/SequentialExecutor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/CharMatcher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/FluentIterable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Booleans.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Chars.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/net/PercentEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Joiner.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Floats.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ListenableFutureTask.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/escape/UnicodeEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Bytes.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/MultiReader.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/base/Stopwatch.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Converter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Longs.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Ints.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/internal/Finalizer.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/net/InetAddresses.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/escape/Escaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/thirdparty/publicsuffix/TrieParser.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/MinMaxPriorityQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/AbstractSequentialIterator.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RowSortedTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingSortedSetMultimap.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingImmutableCollection.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/UnmodifiableListIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SortedMapDifference.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not
* use this file except in compliance with the License. You may obtain a copy of
* the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SortedMultisets.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/collect/DenseImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/UnsignedBytes.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/HostSpecifier.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractExecutionThreadService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/JdkFutureAdapters.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/SignedBytes.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/annotations/GwtIncompatible.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SparseImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/UrlEscapers.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/ReferenceEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/MapMakerInternalMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/TypeResolver.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/Platform.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/MapMaker.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/ArrayBasedUnicodeEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/CacheBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Cut.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ByteArrayDataInput.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/SettableFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Callables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Service.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractIdleService.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingFluentFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Platform.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ByteArrayDataOutput.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/html/HtmlEscapers.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/ArrayBasedEscaperMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/LineProcessor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/xml/XmlEscapers.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingListenableFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/annotations/GwtCompatible.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ByteProcessor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/ArrayBasedCharEscaper.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/InternetDomainName.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Splitter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/LocalCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/Escapers.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software

```
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
  * Not supported. <b>You are attempting to create a map that may contain a non-{@code Comparable}
  * key.</b> Proper calls will resolve to the version in {@code ImmutableSortedMap}, not this dummy
  * version.
  *
  * @throws UnsupportedOperationException always
  * @deprecated <b>Pass a key of type {@code Comparable}
 to use {@link
  *     ImmutableSortedMap#of(Comparable, Object)}.</b>
  */
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedMapFauxverideShim.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Floats}. Intended to be empty for regular
* version.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/FloatsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2009 The Guava Authors
*
```

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ComputationException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/DiscreteDomain.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableClassToInstanceMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedSetFauxverideShim.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableAsList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ArrayTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TableCollectors.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableEnumSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableSortedSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SingletonImmutableList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SingletonImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ComparisonChain.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractIndexedListIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/EmptyImmutableSetMultimap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ContiguousSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Monitor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/package-info.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SortedLists.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingBlockingQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Ascii.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Strings.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/annotations/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Atomics.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Equivalence.java
*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/annotations/Beta.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ThreadFactoryBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ListeningExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/UncaughtExceptionHandlers.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/GwtTransient.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AtomicLongMap.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/PatternCompiler.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/CommonMatcher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/CommonPattern.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/JdkPattern.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/MutableValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/AbstractGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/AbstractUndirectedNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/MoreCollectors.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ImmutableValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ElementOrder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ValueGraphBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/NetworkConnections.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Comparators.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/GraphConstants.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ForwardingGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/AbstractNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/EdgesConnecting.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/GraphBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/UndirectedMultiNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/EndpointPairIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ForwardingValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/AbstractValueGraph.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/EndpointPair.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/DirectedNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableMultisetGwtSerializationDependencies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/StandardMutableNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/UndirectedNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/StandardMutableValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/UndirectedGraphConnections.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ForwardingNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/MultiEdgesConnecting.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RangeGwtSerializationDependencies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/LinkedHashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/collect/ArrayListMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/StandardValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/DirectedGraphConnections.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/StandardNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/CollectCollectors.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/MapRetrievalCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/DirectedMultiNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/GraphConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/AbstractGraphBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/NetworkBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/MapIteratorCache.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/AbstractDirectedNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/StandardMutableGraph.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Doubles}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/DoublesMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Ints}. Intended to be empty for regular
* version.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/IntsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CollectSpliterators.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableBiMapFauxverideShim.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/package-info.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedAsList.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/Resources.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Predicates.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Function.java

*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Throwables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/AsyncEventBus.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Primitives.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/FinalizableWeakReference.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ExecutionList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/DeadEvent.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/Files.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/package-info.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/LineReader.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Predicate.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Preconditions.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/FinalizableSoftReference.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/FinalizableReferenceQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Objects.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Charsets.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/Subscribe.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/MultiInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/Flushables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Supplier.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/FinalizableReference.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Interners.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/Closeables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/eventbus/EventBus.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/FinalizablePhantomReference.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/LineBuffer.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Suppliers.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/DirectExecutor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/EnumMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/eventbus/AllowConcurrentEvents.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/LittleEndianDataInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Defaults.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/CountingOutputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ListenableFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/AbstractIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Functions.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/CountingInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/LittleEndianDataOutputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/CharStreams.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ByteStreams.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/LittleEndianByteArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AsyncCallable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/FarmHashFingerprint64.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/ReaderInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/MacHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ConsumingQueueIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/CombinedFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Platform.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/InterruptibleTask.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AggregateFutureState.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/IncidentEdgeSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CompactHashing.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingMapEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Sets.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Multimaps.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/BiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/collect/Multisets.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingConcurrentMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ReverseNaturalOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SortedSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/NullsFirstOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SingletonImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/EnumHashBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Multiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingListIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Ordering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TreeMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/LinkedListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractSortedSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ConcurrentHashMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/NullsLastOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Multimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Iterators.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Interner.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Synchronized.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ComparatorOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CompoundOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingSet.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractMapBasedMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingSortedMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/MutableClassToInstanceMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractMapBasedMultiset.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/UsingToStringOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Iterables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TreeMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/NaturalOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Lists.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ExplicitOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/LinkedHashMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ArrayListMultimap.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/collect/ByFunctionOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ReverseOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/SetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/AbstractMapEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/LexicographicalOrdering.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/LinkedHashMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/HashMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingCollection.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ClassToInstanceMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Maps.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingSortedSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingObject.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/MapDifference.java
*
/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/EnumBiMap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/ImmutableDoubleArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/ImmutableIntArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingLock.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/AbstractHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/ImmutableLongArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingCondition.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Platform.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Internal.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you
* may not use this file except in compliance with the License.  You may
* obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
* implied.  See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Streams.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/Java8Compatibility.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/OverflowAvoidingLockSupport.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/ToDoubleRounder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/BigDecimalMath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Java8Compatibility.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/IgnoreJRERequirement.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/ImmutableSupplier.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ExecutionSequencer.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0

*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
  * This method was written by Doug Lea with assistance from members of JCP JSR-166 Expert Group
  * and released to the public domain, as explained at
  * http://creativecommons.org/licenses/publicdomain
  *
  * As of 2010/06/11, this method is identical to the (package private) hash method in OpenJDK 7's
  * java.util.HashMap
 class.
  */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Striped.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableSortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSortedMultisetFauxverideShim.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/GeneralRange.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingSortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/collect/SortedIterable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RangeSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/AbstractRangeSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Count.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/SortedIterables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSortedMultiset.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/UnsignedInts.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/WrappingExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractScheduledService.java
*
 /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/UncheckedExecutionException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/Weigher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/CacheStats.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/AbstractCompositeHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/Hashing.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/AbstractLoadingCache.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/ForwardingLoadingCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/CacheBuilderSpec.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Optional.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/DescendingImmutableSortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Ticker.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Present.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/UnsignedInteger.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Murmur3_32HashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/PrimitiveSink.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/BoundType.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/AbstractHasher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/RemovalListeners.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/UnsignedLong.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/Types.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/TypeParameter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/PairwiseEquivalence.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/CycleDetectingLockFactory.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TreeRangeSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularContiguousSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/base/Absent.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/RemovalNotification.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractListeningExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Hasher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingListeningExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/HashCode.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/MediaType.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/AbstractStreamingHasher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Crc32cHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Uninterruptibles.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/HostAndPort.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Murmur3_128HashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Enums.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/MathPreconditions.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/MessageDigestHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/UnsignedLongs.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/RemovalListener.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ListeningScheduledExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/HttpHeaders.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/RemovalCause.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/FunctionalEquivalence.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AsyncFunction.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/Cache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/BloomFilterStrategies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/BigIntegerMath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/AbstractCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ExecutionError.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/IntMath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/HashingOutputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/EmptyContiguousSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/DoubleMath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Funnels.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/CacheLoader.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/AbstractNonStreamingHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Funnel.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/ForwardingCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Queues.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/LongMath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/DoubleUtils.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractSortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/LoadingCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/ParseRequest.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/FutureCallback.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/BloomFilter.java
*

/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/HashFunction.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
  * Returns an array containing all of the elements in the specified collection. This method
  * returns the elements in the order they are returned by the collection's iterator. The returned
  * array is "safe" in that no references to it are maintained by the collection. The caller is
  * thus free to modify the returned
 array.
  *
  * <p>This method assumes that the collection size doesn't change while the method is running.
  *
  * <p>TODO(kevinb): support concurrently modified collections?
  *
  * @param c the collection for which to return an array of elements
  */
```

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ObjectArrays.java

# 1.70 jackson-datatype-threetenbp 2.10.0

## 1.70.1 Available under license :

This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# 1.71 proto-google-common-protos 2.0.1

## 1.71.1 Available under license :

No license file was found, but licenses were detected in source scan.

// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//  http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/http.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/context.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/quota.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/monitored_resource.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/type/datetime.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/system_parameter.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/longrunning/operations.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/type/expr.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/logging/type/http_request.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/endpoint.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/rpc/context/attribute_context.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/httpbody.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/type/quaternion.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/annotations.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/usage.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/backend.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/monitoring.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/consumer.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-

jar/google/api/field_behavior.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/control.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/metric.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/log.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/cloud/audit/audit_log.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/type/color.proto
*
/opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/rpc/code.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/api/distribution.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/label.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/service.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/type/fraction.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/type/postal_address.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/type/latlng.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/geo/type/viewport.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/api/source_info.proto
*
/opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/logging.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/billing.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/logging/type/log_severity.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/type/calendar_period.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/rpc/error_details.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/api/config_change.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/type/dayofweek.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/type/date.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/type/money.proto
*
/opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/type/timeofday.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/api/resource.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-
jar/google/api/documentation.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/rpc/status.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/auth.proto
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/client.proto

No license file was found, but licenses were detected in source scan.

// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//  http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,
// agreement (which includes confidentiality provisions). These features may

Found in path(s):
* /opt/cola/permits/1127519988_1611825907.91/0/proto-google-common-protos-2-0-1-jar/google/api/launch_stage.proto

# 1.72 commons-io 2.6

## 1.72.1 Available under license :

Apache Commons IO
Copyright 2002-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.73 checker-compat-qual 2.5.5

## 1.73.1 Available under license :

```
/*
* $Header: /home/cvs/jakarta-commons/LICENSE,v 1.4 2002/04/11 13:24:02 dion Exp $
* $Revision: 1.4 $
```

```
* $Date: 2002/04/11 13:24:02 $
*
* ====================================================================
*
* The Apache Software License, Version 1.1
*
* Copyright (c) 1999-2001 The Apache Software Foundation.  All rights
* reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
*
* 3. The end-user documentation included with the redistribution, if
*    any, must include the following acknowlegement:
*
      "This product includes software developed by the
*       Apache Software Foundation (http://www.apache.org/)."
*    Alternately, this acknowlegement may appear in the software itself,
*    if and wherever such third-party acknowlegements normally appear.
*
* 4. The names "The Jakarta Project", "Commons", and "Apache Software
*    Foundation" must not be used to endorse or promote products derived
*    from this software without prior written permission. For written
*    permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache"
*    nor may "Apache" appear in their names without prior written
*    permission of the Apache Group.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS
 BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
```

* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* ==========
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*
*/

Most of the Checker Framework is licensed under the GNU General Public License, version 2 (GPL2), with the classpath exception. The text of this license appears below. This is the same license used for OpenJDK.

A few parts of the Checker Framework have more permissive licenses.

* The annotations are licensed under the MIT License. (The text of this license appears below.) More specifically, all the parts of the Checker Framework that you might want to include with your own program use the MIT License. This is the checker-qual.jar file and all the files that appear in it: every file in a qual/ directory, plus utility files such as NullnessUtil.java, RegexUtil.java, SignednessUtil.java, etc. In addition, the cleanroom implementations of third-party annotations, which the Checker Framework recognizes as aliases for its own annotations, are licensed under the MIT License.

Some external libraries that are included with the Checker Framework have different licenses.

* javaparser is dual licensed under the LGPL or the Apache license -- you may use it under whichever one you want. (The javaparser source code contains a file with the text of the GPL, but it is not clear why, since javaparser does not use the GPL.) See file stubparser/LICENSE and the source code of all its files.

* JUnit is licensed under the Common Public License v1.0 (see http://www.junit.org/license), with parts (Hamcrest) licensed under the BSD License (see http://hamcrest.org/JavaHamcrest/).

* Libraries in plume-lib (https://github.com/plume-lib/) are licensed under the MIT License.

The Checker Framework includes annotations for the JDK in directory checker/jdk/, and for some other libraries. Each annotated library uses the same license as the unannotated version of the library.

==========

The GNU General Public License (GPL)

Version 2, June 1991

Copyright (C) 1989, 1991 Free
Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license
document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share
and change it. By contrast, the GNU General Public License is intended to
guarantee your freedom to share and change free software--to make sure the
software is free for all its users. This General Public License applies to
most of the Free Software Foundation's software and to any other program whose
authors commit to using it. (Some other Free Software Foundation software is
covered by the GNU Library General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not price. Our
General Public Licenses are designed to make sure that you have the freedom to
distribute copies of free software (and charge for this service if
you wish),
that you receive source code or can get it if you want it, that you can change
the software or use pieces of it in new free programs; and that you know you
can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny
you these rights or to ask you to surrender the rights. These restrictions
translate to certain responsibilities for you if you distribute copies of the
software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for
a fee, you must give the recipients all the rights that you have. You must
make sure that they, too, receive or can get the source code. And you must
show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2)
offer you this license which gives you legal permission to copy, distribute
and/or modify the software.

Also, for each author's protection and ours, we want to make certain that
everyone understands

that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it.
For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program

subject to these terms and conditions.
You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely
from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If
the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation.
If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE
PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want
it to be of the greatest possible
use to the public, the best way to achieve this is to make it free software
which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever
suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL

Certain source files distributed by Oracle America and/or its affiliates are subject to the following clarification and special exception to the GPL, but only where Oracle has expressly included
in the particular source file's header
the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the LICENSE file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend
this exception to your version of
the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

==========

MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
==========================================================================
/*
* $Header: /home/cvs/jakarta-struts/LICENSE,v 1.2 2001/02/02 00:38:31 craigmcc Exp $
* $Revision: 1.2 $
* $Date: 2001/02/02 00:38:31 $
*
* ====================================================================
*
* The Apache Software License, Version 1.1
*
* Copyright (c) 1999-2001 The Apache Software Foundation.  All rights
* reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
*
* 3. The end-user documentation included with the redistribution, if
*    any, must include the
 following acknowlegement:
*       "This product includes software developed by the
*        Apache Software Foundation (http://www.apache.org/)."
*    Alternately, this acknowlegement may appear in the software itself,
*    if and wherever such third-party acknowlegements normally appear.
*
* 4. The names "The Jakarta Project", "Struts", and "Apache Software
*    Foundation" must not be used to endorse or promote products derived
*    from this software without prior written permission. For written
*    permission, please contact apache@apache.org.
*
```

```
* 5. Products derived from this software may not be called "Apache"
*    nor may "Apache" appear in their names without prior written
*    permission of the Apache Group.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED.  IN NO EVENT SHALL
 THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* ====================================================================
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation.  For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*
*/
/*
* $Header: /home/cvspublic/jakarta-commons/logging/LICENSE.txt,v 1.2 2003/04/06 20:37:31 rdonkin Exp $
* $Revision: 1.2 $
* $Date: 2003/04/06 20:37:31 $
*
* ====================================================================
*
* The Apache Software License, Version 1.1
*
* Copyright (c) 1999-2003 The Apache Software Foundation.  All rights
* reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
```

*
* 3. The end-user documentation included with the redistribution, if
*    any, must include the following
 acknowlegement:
*       "This product includes software developed by the
*        Apache Software Foundation (http://www.apache.org/)."
*    Alternately, this acknowlegement may appear in the software itself,
*    if and wherever such third-party acknowlegements normally appear.
*
* 4. The names "The Jakarta Project", "Commons", and "Apache Software
*    Foundation" must not be used to endorse or promote products derived
*    from this software without prior written permission. For written
*    permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache"
*    nor may "Apache" appear in their names without prior written
*    permission of the Apache Group.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION
 OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* ====================================================================
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation.  For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*
*/
   GNU LESSER GENERAL PUBLIC LICENSE
       Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
   59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts

as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest
you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients
all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that
there is no warranty for the free library. Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original

author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU

operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without
limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does

and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves,

then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to
a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed
under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot

use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues),

conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute
so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people
have made
generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License
from time to time.
Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and

conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of
all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL
DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.74 annotations 13.0

## 1.74.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2006 Sascha Weinreuter
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
```

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/Subst.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/Pattern.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/Language.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/PrintFormat.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/RegExp.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/Identifier.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2000-2012 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/jetbrains/annotations/TestOnly.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/MagicConstant.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/JdkConstants.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/jetbrains/annotations/NotNull.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2000-2009 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/jetbrains/annotations/NonNls.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/jetbrains/annotations/Nullable.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/jetbrains/annotations/Nls.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/jetbrains/annotations/PropertyKey.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2000-2013 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/Flow.java

* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-jar/org/jetbrains/annotations/Contract.java

# 1.75 rgxgen 1.3

## 1.75.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2019 Vladislavs Varslavans

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.76 mailapi 1.6.2

## 1.76.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation,
method, process,
and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in

combination with its Contributor
Version (or portions of such
combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software
in the absence of
Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained

within the Covered Software, or any notices of licensing or any
descriptive text giving attribution to any Contributor or the
Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in
Source Code form that alters or restricts the applicable version of
this License or the recipients' rights hereunder. You may choose to
offer, and to charge a fee for, warranty, support, indemnity or
liability obligations to one or more recipients of Covered Software.
However, you may do so only on Your own behalf, and not on behalf of
the Initial Developer or any Contributor. You must make
it
absolutely clear that any such warranty, support, indemnity or
liability obligation is offered by You alone, and You hereby agree
to indemnify the Initial Developer and every Contributor for any
liability incurred by the Initial Developer or such Contributor as a
result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under
the terms of this License or under the terms of a license of Your
choice, which may contain terms different from this License,
provided that You are in compliance with the terms of this License
and that the license for the Executable form does not attempt to
limit or alter the recipient's rights in the Source Code form from
the rights set forth in this License. If You distribute the Covered
Software in Executable form under a different license, You must make
it absolutely clear that any
terms which differ from this License
are offered by You alone, not by the Initial Developer or
Contributor. You hereby agree to indemnify the Initial Developer and
every Contributor for any liability incurred by the Initial
Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with
other code not governed by the terms of this License and distribute
the Larger Work as a single product. In such a case, You must make
sure the requirements of this License are fulfilled for the Covered
Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License
if You: (a) rename the license and
remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED
HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR

CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control

laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

---

NOTICE PURSUANT TO SECTION
9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION
LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.
This General Public
License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either

verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If

identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is

normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs,
unless that component itself accompanies the
executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works.
These actions are
prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License,
they do not
excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you

could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we

sometimes make exceptions for this. Our decision will be
guided by the
two goals of preserving the free status of all derivatives of our free
software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND,
EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH
YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL
NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR
DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL
DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM
(INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED
INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF
THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR
OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to
attach them to the start of each source file to most effectively convey
the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief
idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or

(at your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version
69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the commands
you use may be called something other than `show w' and `show c'; they
could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by
James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your
program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications
with the library. If this is what you want to do, use the GNU Library
General Public License instead of this License.

#

Certain source files distributed by Oracle America, Inc. and/or its

affiliates are subject to the following clarification and special exception to the GPLv2, based on the GNU Project exception for its Classpath libraries, known as the GNU Classpath Exception, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the LICENSE file that accompanied this code."

You should also note that Oracle includes multiple, independent programs in this software package. Some of those programs are provided under licenses deemed incompatible with the GPLv2 by the Free Software Foundation and
others. For example, the package includes programs licensed under the Apache License, Version 2.0. Such programs are licensed to you under their original licenses.

Oracle facilitates your further distribution of this package by adding the Classpath Exception to the necessary parts of its GPLv2 code, which permits you to use that code in combination with other independent modules not licensed under the GPLv2. However, note that this would not permit you to commingle code under an incompatible license with Oracle's GPLv2 licensed code by, for example, cutting and pasting such code into a file also containing Oracle's GPLv2 licensed code and then distributing the result. Additionally, if you were to remove the Classpath Exception from any of the files to which it applies and distribute the result, you would likely be required to license some or all of the other code in that distribution under the GPLv2 as well, and since the GPLv2 is incompatible with the license terms of some items included
in the distribution by Oracle, removing the Classpath Exception could therefore effectively compromise your ability to further distribute the package.

Proceed with caution and we recommend that you obtain the advice of a lawyer skilled in open source matters before removing the Classpath Exception or making modifications to this package which may subsequently be redistributed and/or involve the use of third party software.

CLASSPATH EXCEPTION
Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent

modules, and to copy and distribute the resulting executable under
terms of your choice, provided that you also meet,
for each linked
independent module, the terms and conditions of the license of that
module. An independent module is a module which is not derived from or
based on this library. If you modify this library, you may extend this
exception to your version of the library, but you are not obligated to
do so. If you do not wish to do so, delete this exception statement
from your version.

# 1.77 libphonenumber 8.11.1

## 1.77.1 Available under license :

Apache Commons IO
Copyright 2001-2008 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

This project contains a small number of files derived from the Chromium
project. They contain copyright notices such as this:

// Copyright (c) 2010 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

Here is the full text of the LICENSE file from the Chromium project:

// Copyright 2014 The Chromium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors
may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (C) 2011 The Libphonenumber Authors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
# This is the list of people who have contributed to the repository.
# The AUTHORS file lists the copyright holders; this file
# lists people.  For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.

# Core team:
#
# These people were responsible for the initial implementations in Java and
# C++.

Shaopeng Jia
Lara Rennie
David Yonge-Mallo
David Beaumont

# Team carrying out the bulk of metadata maintenance, bug fixes & feature
# requests, and porting new features to Javascript.

Andy Staudacher
Aravind Penmetsa
David Beaumont
Kavitha Keshava
Keghani Kristelle Kouzoujian
Sumathi Sundara Murthy

# Additional contributions:

# Javascript
Nikolaos Tronikos
  -- Initial implementation, some subsequent maintenance.
James Wright
  -- Short number info port to Javascript.

# Java:
Walter Erqunigo
  -- Timezone mapper.
Philippe Liard
  -- Offline geocoder, local Maven expert.
Cecilia Roes
  -- Carrier mapper.
Tom Hofmann
  -- First version
 of the PhoneNumberMatcher and related classes.
Fabien Allanic
Kushal Pisavadia

# C++:
Patrick Mzard
  -- Geocoding library implementation.
Philippe Liard
  -- Initial open-sourcing of a C++ library, responsible for getting this to
     work across multiple platforms.
Ben Gertzfield
Fredrik Roubert
George Yakovlev
Nico Weber
Yoshifumi Inoue

# Debian:
Fredrik Roubert

# Metadata:
Alec McTurk
Alexandru Manea
Amir Farsi
Andrs Eisenberger
Anssi Hannula
Boshi Lian -- Maintains Chinese geocoding data
Bradford Smith
Cecilia Roes
Florian Grisch
Fredrik Roubert
Ian Galpin -- Maintains the data for the UK
Javier Kohen

Jeanine Lilleng
Karen Lees
Khaled Hafez
Lei Zhang
monjiwired@
Philippe Liard
Sang-il Lee
Yugui Sonoda

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or
(iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from)
the Work and for which the
editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication
on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant
of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as
of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own
copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

BSD License

Copyright (c) 2000-2006, www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer. Redistributions in binary form must reproduce
the above copyright notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse
or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY

EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL
THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY
WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Commons FileUpload
Copyright 2002-2008 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache Commons Lang
Copyright 2001-2011 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: libphonenumber
Upstream-Contact: Fredrik Roubert <roubert@google.com>
Source: https://github.com/google/libphonenumber/

Files: * cpp/src/phonenumbers/base/logging.h

cpp/src/phonenumbers/base/thread_checker.h
cpp/src/phonenumbers/base/memory/singleton.h
cpp/src/phonenumbers/base/memory/singleton_posix.h
cpp/src/phonenumbers/base/synchronization/lock.h
cpp/src/phonenumbers/base/synchronization/lock_posix.h
cpp/src/phonenumbers/base/thread_checker.h
Copyright: 2011, The Libphonenumber Authors
License: Apache-2.0
On Debian systems, the full text of the Apache
License version 2.0 can be found in the file
`/usr/share/common-licenses/Apache-2.0'.

Files: cpp/src/phonenumbers/utf/rune.c
cpp/src/phonenumbers/utf/utf.h
cpp/src/phonenumbers/utf/utfdef.h
Copyright: 1998-2002, Lucent Technologies
License: MIT

Files: cpp/src/phonenumbers/base/*
tools/cpp/src/base/*
Copyright: 2006-2012, The Chromium
 Authors
License: BSD

Files: debian/*
Copyright: 2011, The Libphonenumber Authors
 2014, Daniel Pocock http://danielpocock.com
License: Apache-2.0
On Debian systems, the full text of the Apache
License version 2.0 can be found in the file
`/usr/share/common-licenses/Apache-2.0'.

License: MIT
Permission to use, copy, modify, and distribute this software for any
purpose without fee is hereby granted, provided that this entire notice
is included in all copies of any software which is or includes a copy
or modification of this software and in all copies of the supporting
documentation for such software.
THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED
WARRANTY. IN PARTICULAR, NEITHER THE AUTHORS NOR LUCENT TECHNOLOGIES MAKE ANY
REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY
OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

License: BSD
Redistribution and use in source and binary forms, with or without
modification,
 are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.78 jctools-core 3.1.0

## 1.78.1 Available under license :

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred
form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the

interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal
Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of
their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution
notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that
such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor
be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted
against, such Contributor by reason of your accepting any such warranty or additional liability.
END OF TERMS AND CONDITIONS

# 1.79 commons-lang3 3.8.1

## 1.79.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and
limitations under the License.
Apache Commons Lang
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

# 1.80 commons-codec 1.11

## 1.80.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or
(iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as
of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own
copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names
of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law
(such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your

sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Commons Codec
Copyright 2002-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

==========

The content of package org.apache.commons.codec.language.bm has been translated
from the original php source code available at http://stevemorse.org/phoneticinfo.htm
with permission from the original authors.

Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

# 1.81 commons-cli 1.2

## 1.81.1 Available under license :

Apache Commons CLI
Copyright 2001-2009 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.82 swagger-annotations 2.1.7

## 1.82.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/**
* Copyright 2017 SmartBear Software
* <p>
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* <p>
* http://www.apache.org/licenses/LICENSE-2.0
* <p>
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/media/Encoding.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/tags/Tags.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/tags/Tag.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/callbacks/Callback.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/media/DiscriminatorMapping.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/ExternalDocumentation.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/info/Info.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/security/OAuthScope.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/media/Content.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/Operation.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/Hidden.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/servers/Server.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/security/SecurityRequirement.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/security/SecurityScheme.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/security/SecurityRequirements.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/links/LinkParameter.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/extensions/Extensions.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/security/OAuthFlow.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/links/Link.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/media/Schema.java

* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/servers/ServerVariable.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/headers/Header.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/Parameter.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/callbacks/Callbacks.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/responses/ApiResponse.java
*
 /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/security/SecuritySchemes.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/OpenAPIDefinition.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/responses/ApiResponses.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/info/License.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/parameters/RequestBody.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/media/ExampleObject.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/security/OAuthFlows.java
*
 /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/info/Contact.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/Parameters.java
* /opt/cola/permits/1152463053_1618242054.03/0/swagger-annotations-2-1-7-sources-jar/io/swagger/v3/oas/annotations/servers/Servers.java

# 1.83 swagger-models 2.1.7

## 1.83.1 Available under license :

No license file was found, but licenses were detected in source scan.

/**
* Copyright 2017 SmartBear Software
* <p>
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* <p>
* http://www.apache.org/licenses/LICENSE-2.0
* <p>

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/servers/ServerVariable.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/responses/ApiResponses.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/info/Info.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/parameters/PathParameter.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/links/LinkParameter.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/Paths.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/BooleanSchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/callbacks/Callback.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/DateTimeSchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/Operation.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/OpenAPI.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/responses/ApiResponse.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/tags/Tag.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/FileSchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/Encoding.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/PathItem.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/info/Contact.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/ComposedSchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/parameters/CookieParameter.java

*
/opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/Schema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/parameters/HeaderParameter.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/security/SecurityRequirement.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/security/SecurityScheme.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/Content.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/MediaType.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/EncodingProperty.java
*
/opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/examples/Example.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/BinarySchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/ExternalDocumentation.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/ObjectSchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/ArraySchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/UUIDSchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/ByteArraySchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/MapSchema.java
*
/opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/DateSchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/servers/Server.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/headers/Header.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/Components.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/XML.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/servers/ServerVariables.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/IntegerSchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-

jar/io/swagger/v3/oas/models/media/EmailSchema.java
*
/opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/links/Link.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/StringSchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/PasswordSchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/security/OAuthFlows.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/parameters/Parameter.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/info/License.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/parameters/RequestBody.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/security/Scopes.java
*
/opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/parameters/QueryParameter.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/media/NumberSchema.java
* /opt/cola/permits/1152463019_1618242042.83/0/swagger-models-2-1-7-sources-jar/io/swagger/v3/oas/models/security/OAuthFlow.java

# 1.84 jackson-coreutils-equivalence 1.0

## 1.84.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2014, Francis Galiegue (fgaliegue@gmail.com)
*
* This software is dual-licensed under:
*
* - the Lesser General Public License (LGPL) version 3.0 or, at your option, any
*   later version;
* - the Apache Software License (ASL) version 2.0.
*
* The text of this file and of both licenses is available at the root of this
* project or, if you have the jar distribution, in directory META-INF/, under
* the names LGPL-3.0.txt and ASL-2.0.txt respectively.
*
* Direct link to the sources:
*
* - LGPL 3.0: https://www.gnu.org/licenses/lgpl-3.0.txt
```

* - ASL 2.0: http://www.apache.org/licenses/LICENSE-2.0.txt
*/

Found in path(s):
* /opt/cola/permits/1166302096_1621381491.42/0/jackson-coreutils-equivalence-1-0-sources-jar/com/github/fge/jackson/JsonNumEquivalence.java

# 1.85 json-schema-core 1.2.14

## 1.85.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by
the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters,
data structure
layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse

engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.
This software is dual-licensed under:

- the Lesser General Public License (LGPL) version 3.0 or, at your option, any later version;
- the Apache Software License (ASL) version 2.0.

The text of both licenses is included (under the names LGPL-3.0.txt and ASL-2.0.txt respectively).

Direct link to the sources:

- LGPL 3.0: https://www.gnu.org/licenses/lgpl-3.0.txt
- ASL 2.0: http://www.apache.org/licenses/LICENSE-2.0.txt

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

## Fabio Di Fabio

* Allow to disable schema caching (except for preloaded schemas)

## Randy Watler

* Support for reading schemas from non standard JSON (comments etc).

## Matt Bishop

* OSGi support for Maven (now deprecated).

# 1.86 uritemplate 0.10

## 1.86.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public

License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs

whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of
this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from
a header file that is part of the Library. You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters,
data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the
Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license
document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

a) Give prominent notice with each copy of the Combined Work that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license
document.

c) For a Combined Work that
displays copyright notices during
execution, include the copyright notice for the Library among
these notices, as well as a reference directing the user to the
copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this
License, and the Corresponding Application Code in a form
suitable for, and under terms that permit, the user to

recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version
of the GNU Lesser General Public License "or any later version"
applies to it, you have the option
of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.
This software is dual-licensed under:

- the Lesser General Public License (LGPL) version 3.0 or, at your option, any
later version;
- the Apache Software License (ASL) version 2.0.

The text of both licenses is included (under the names LGPL-3.0.txt and
ASL-2.0.txt respectively).

Direct link to the sources:

- LGPL 3.0: https://www.gnu.org/licenses/lgpl-3.0.txt
- ASL 2.0: http://www.apache.org/licenses/LICENSE-2.0.txt

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.87 json-schema-validator 2.2.14

## 1.87.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by
the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters,
data structure
layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based
on the Library, uncombined with any other library facilities,
conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it
is a work based on the Library, and explaining where to find the
accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions
of the GNU Lesser General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version
of the GNU Lesser General Public License "or any later version"
applies to it, you have the option
of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.
This software is dual-licensed under:

- the Lesser General Public License (LGPL) version 3.0 or, at your option, any
later version;
- the Apache Software License (ASL) version 2.0.

The text of both licenses is included (under the names LGPL-3.0.txt and
ASL-2.0.txt respectively).

Direct link to the sources:

- LGPL 3.0: https://www.gnu.org/licenses/lgpl-3.0.txt
- ASL 2.0: http://www.apache.org/licenses/LICENSE-2.0.txt
### Pablo Pinhero

* Add a "lib" target to the build: jar with all dependencies builtin.

### Corey Sciuto

* Tests for date-time format attribute; date-time-millis format attribute (now superseded by date-time itself).

### Matt Bishop:

* Initial OSGi support for maven (now superseded by Gradle).

### Kelvin Pho

* Pointer collectors for draft v4 keywords (since then moved to -core).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.88 asm 9.1

## 1.88.1 Available under license :

No license file was found, but licenses were detected in source scan.

// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
// notice, this list of conditions and the following disclaimer in the
// documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its

// this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/ModuleWriter.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/FieldWriter.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/ClassWriter.java
*
/opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/SymbolTable.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/TypeReference.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/Edge.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-
jar/org/objectweb/asm/RecordComponentVisitor.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-
jar/org/objectweb/asm/signature/SignatureWriter.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/FieldVisitor.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/Opcodes.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/AnnotationWriter.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/ClassReader.java
*
/opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/Handler.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-
jar/org/objectweb/asm/RecordComponentWriter.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/ByteVector.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-
jar/org/objectweb/asm/AnnotationVisitor.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/TypePath.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-
jar/org/objectweb/asm/MethodTooLargeException.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-
jar/org/objectweb/asm/signature/SignatureReader.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/CurrentFrame.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/ConstantDynamic.java
*
/opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/MethodVisitor.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-
jar/org/objectweb/asm/signature/SignatureVisitor.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-
jar/org/objectweb/asm/ClassTooLargeException.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/ClassVisitor.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/Label.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/Type.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/Constants.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/ModuleVisitor.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/MethodWriter.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/Frame.java
*
/opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/Context.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/Attribute.java

* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/Handle.java
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/Symbol.java
No license file was found, but licenses were detected in source scan.

2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*  notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*  notice, this list of conditions and the following disclaimer in the
*  documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*  contributors may be used to endorse or promote products derived from
*  this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/signature/package.html
* /opt/cola/permits/1175494765_1624391692.09/0/asm-9-1-sources-2-jar/org/objectweb/asm/package.html

# 1.89 swagger-core 2.1.7

## 1.89.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2020 SmartBear Software Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.90 gson 2.8.6

## 1.90.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2010 The Android Open Source Project
* Copyright (C) 2012 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/LinkedTreeMap.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/LinkedHashTreeMap.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
```

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/CollectionTypeAdapterFactory.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/TypeAdapters.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/JsonTreeReader.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/ArrayTypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/ObjectTypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/MapTypeAdapterFactory.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/TypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/JsonReaderInternalAccess.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/JsonTreeWriter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/TypeAdapterFactory.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/ReflectiveTypeAdapterFactory.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/TypeAdapterRuntimeTypeWrapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Gson authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/GsonBuildConfig.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Gson authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/reflect/PreJava9ReflectionAccessor.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/PreJava9DateFormatProvider.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/reflect/ReflectionAccessor.java
*
 /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/reflect/UnsafeReflectionAccessor.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/JavaVersion.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonParseException.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonPrimitive.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/ExclusionStrategy.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonDeserializer.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/$Gson$Preconditions.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/Excluder.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonSerializationContext.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonArray.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/annotations/Since.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/Primitives.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/reflect/TypeToken.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/Gson.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonIOException.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonObject.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonSerializer.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/ObjectConstructor.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonNull.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonDeserializationContext.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonElement.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/GsonBuilder.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/FieldNamingPolicy.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/FieldNamingStrategy.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/DefaultDateTypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-

jar/com/google/gson/annotations/Until.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/annotations/SerializedName.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/annotations/Expose.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/InstanceCreator.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2014 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/annotations/JsonAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/internal/bind/JsonAdapterAnnotationTypeAdapterFactory.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/LazilyParsedNumber.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/SqlDateTypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/ConstructorConstructor.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/TimeTypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/DateTypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/UnsafeAllocator.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/TreeTypeAdapter.java
No license file was found, but licenses were detected in source scan.

```
/**
* Copyright (C) 2008 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/$Gson$Types.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2010 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
```

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonSyntaxException.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/Streams.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/stream/JsonToken.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/stream/JsonWriter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/stream/JsonScope.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/stream/MalformedJsonException.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/stream/JsonReader.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at

*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/LongSerializationPolicy.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonStreamParser.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/FieldAttributes.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonParser.java

# 1.91 xsdlib 2013.6.1

## 1.91.1 Available under license :

No license file was found, but licenses were detected in source scan.

# Copyright (c) 2001-2013 Oracle and/or its affiliates. All rights reserved.
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met:
#  - Redistributions of source code must retain the above copyright
#    notice, this list of conditions and the following disclaimer.
#  - Redistributions in binary form must reproduce the above copyright
#    notice, this list of conditions and the following disclaimer in the
#    documentation and/or other materials provided with the distribution.
#  - Neither the name of Oracle nor the names of its
#    from this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/Messages.properties
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/Messages_ja.properties
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2001-2013 Oracle and/or its affiliates. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions
* are met:
*
*  - Redistributions of source code must retain the above copyright
*  notice, this list of conditions and the following disclaimer.
*
*  - Redistributions in binary form must reproduce the above copyright
*  notice, this list of conditions and the following disclaimer in the
*  documentation and/or other materials provided with the distribution.
*
*  - Neither the name of Oracle nor the names of its
*  contributors may be used to endorse or promote products derived
*  from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
* IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
 BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-
jar/com/sun/msv/datatype/xsd/ShortType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-
jar/com/sun/msv/datatype/xsd/datetime/DateTimeFactory.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-
jar/com/sun/msv/datatype/xsd/datetime/BigTimeDurationValueType.java
*
 /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-
jar/com/sun/msv/datatype/xsd/NonPositiveIntegerType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-
jar/com/sun/msv/datatype/xsd/MinInclusiveFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-
jar/com/sun/msv/datatype/xsd/SimpleURType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-
jar/com/sun/msv/datatype/xsd/XSDatatype.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-
jar/com/sun/msv/datatype/xsd/WhiteSpaceFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-
jar/com/sun/msv/datatype/SerializationContext.java

* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/XSDatatypeImpl.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/DataTypeWithLexicalConstraintFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/UnsignedLongType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/Base64BinaryType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/IDType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/GYearMonthType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/IntegerType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/GDayType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/ListValueType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/DurationType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/ConcreteType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/TokenType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/DataTypeWithFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/AbstractCalendarParser.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/NmtokenType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/CalendarFormatter.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/Proxy.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/CalendarParser.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/TimeZone.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/TotalDigitsFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/regex/RegExp.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/FinalComponent.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-

jar/com/sun/msv/datatype/xsd/GMonthType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/GMonthDayType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/PositiveIntegerType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/DateTimeBaseType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/UnsignedByteType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/BooleanType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/LengthFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/IntType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/GYearType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/LongType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/IDateTimeValueType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/EnumerationFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/AbstractCalendarFormatter.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/DatabindableDatatype.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/DateTimeType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/UnicodeUtil.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/regex/XercesImpl.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/BinaryBaseType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/BuiltinAtomicType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/AnyURIType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/PreciseCalendarFormatter.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/NcnameType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/UnsignedShortType.java

* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/NegativeIntegerType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/MaxExclusiveFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/Discrete.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/IntegerDerivedType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/regex/JDKImpl.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/UnionType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/Comparator.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/PreciseCalendarParser.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/NormalizedStringType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/QnameValueType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/xml/util/XmlChars.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/BigDateTimeValueType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/XmlNames.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/HexBinaryType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/PatternFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/DataTypeWithValueConstraintFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/ngimpl/DatatypeBuilderImpl.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/UnsignedIntType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/TypeIncubator.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/FloatingNumberType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/IntegerValueType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/ITimeDurationValueType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/RangeFacet.java

* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/NameType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/ListType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/TimeType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/MinExclusiveFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/StringType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/ErrorType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/NonNegativeIntegerType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/ErrorDatatypeLibrary.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/FloatType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/WhiteSpaceProcessor.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/DatatypeFactory.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/MinLengthFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/FractionDigitsFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/IDREFType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/regex/RegExpFactory.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/QnameType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/MaxLengthFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/BinaryValueType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/ngimpl/DataTypeLibraryImpl.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/Util.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/NumberType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/DateType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/DoubleType.java

*
/opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/LanguageType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/MaxInclusiveFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/ByteType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/EntityType.java
No license file was found, but licenses were detected in source scan.

2013 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
Neither the name of Oracle nor the names of its
contributors may be used to endorse or promote products derived
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/package.html
No license file was found, but licenses were detected in source scan.

/*
* The Apache Software License, Version 1.1
*
*
* Copyright (c) 1999-2003 The Apache Software Foundation. All rights

* reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
*
* 3. The end-user documentation included with the redistribution,
*    if any, must include the following acknowledgment:
*       "This product includes software developed by the
*        Apache Software Foundation (http://www.apache.org/)."
*    Alternately,
 this acknowledgment may appear in the software itself,
*    if and wherever such third-party acknowledgments normally appear.
*
* 4. The names "Xerces" and "Apache Software Foundation" must
*    not be used to endorse or promote products derived from this
*    software without prior written permission. For written
*    permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache",
*    nor may "Apache" appear in their name, without prior written
*    permission of the Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT
 OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* ====================================================================
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation and was

* originally based on software copyright (c) 1999, International
* Business Machines, Inc., http://www.apache.org.  For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*/

Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/ParserForXMLSchema.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/RegexParser.java
No license file was found, but licenses were detected in source scan.

# Copyright (c) 2001-2013 Oracle and/or its affiliates. All rights reserved.
# modification, are permitted provided that the following conditions are
# - Redistributions  of  source code  must  retain  the above  copyright
#   notice, this list of conditions and the following disclaimer.
#   notice, this list of conditions  and the following disclaimer in the
#   documentation and/or other materials provided with the distribution.
# this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/message_fr.properties
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/message_ja.properties
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/message.properties
No license file was found, but licenses were detected in source scan.

/*
* The Apache Software License, Version 1.1
*
*
* Copyright (c) 1999-2013 The Apache Software Foundation.  All rights
* reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.

*
* 3. The end-user documentation included with the redistribution,
*    if any, must include the following acknowledgment:
*       "This product includes software developed by the
*        Apache Software Foundation (http://www.apache.org/)."
*    Alternately,
 this acknowledgment may appear in the software itself,
*    if and wherever such third-party acknowledgments normally appear.
*
* 4. The names "Xerces" and "Apache Software Foundation" must
*    not be used to endorse or promote products derived from this
*    software without prior written permission. For written
*    permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache",
*    nor may "Apache" appear in their name, without prior written
*    permission of the Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT
 OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* ====================================================================
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation and was
* originally based on software copyright (c) 1999, International
* Business Machines, Inc., http://www.apache.org.  For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*/

Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/RangeToken.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

*
* Copyright (c) 2001-2013 Oracle and/or its affiliates. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are
* met:
*
* - Redistributions of source code must retain the above copyright
*  notice, this list of conditions and the following disclaimer.
*
* - Redistribution in binary form must reproduct the above copyright
*  notice, this list of conditions and the following disclaimer in the
*  documentation and/or other materials provided with the distribution.
*
* Neither the name of Sun Microsystems, Inc. or the names of
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* This software is provided "AS IS," without
 a warranty of any kind. ALL
* EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES,
* INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A
* PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN AND
* ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES OR LIABILITIES
* SUFFERED BY LICENSEE AS A RESULT OF OR RELATING TO USE, MODIFICATION
* OR DISTRIBUTION OF THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL
* SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA,
* OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR
* PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF
* LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE,
* EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*/

Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-
jar/com/sun/msv/datatype/regexp/InternalImpl.java
No license file was found, but licenses were detected in source scan.

/*
* The Apache Software License, Version 1.1
*
*
* Copyright (c) 1999-2002 The Apache Software Foundation. All rights
* reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

```
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
*
* 3. The end-user documentation included with the redistribution,
*    if any, must include the following acknowledgment:
*       "This product includes software developed by the
*        Apache Software Foundation (http://www.apache.org/)."
*    Alternately,
 this acknowledgment may appear in the software itself,
*    if and wherever such third-party acknowledgments normally appear.
*
* 4. The names "Xerces" and "Apache Software Foundation" must
*    not be used to endorse or promote products derived from this
*    software without prior written permission. For written
*    permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache",
*    nor may "Apache" appear in their name, without prior written
*    permission of the Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT
 OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* ====================================================================
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation and was
* originally based on software copyright (c) 1999, International
* Business Machines, Inc., http://www.apache.org.  For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*/
```

Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/Token.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/ParseException.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/RegularExpression.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/Match.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/Op.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/REUtil.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/BMPattern.java

# 1.92 jopt-simple 5.0.4

## 1.92.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
The MIT License

Copyright (c) 2004-2014 Paul R. Holser, Jr.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

*/

Found in path(s):
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/util/EnumConverter.java
No license file was found, but licenses were detected in source scan.

/*
The MIT License

Copyright (c) 2004-2015 Paul R. Holser, Jr.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR
 OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/AlternativeLongOptionSpec.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/OptionParserState.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/UnconfiguredOptionException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/AbstractOptionSpec.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/OptionSpec.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/internal/MethodInvokingValueConverter.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/OptionalArgumentOptionSpec.java
*

/opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/util/DateConverter.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/util/RegexMatcher.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/internal/ConstructorInvokingValueConverter.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/OptionArgumentConversionException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/internal/Classes.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/UnavailableOptionException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/OptionDescriptor.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/ValueConversionException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/OptionParser.java
*
/opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/internal/Reflection.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/UnrecognizedOptionException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/internal/Messages.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/NoArgumentOptionSpec.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/ParserRules.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/RequiredArgumentOptionSpec.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/ArgumentList.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/ValueConverter.java
*
/opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/internal/SimpleOptionNameMap.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/MultipleArgumentsForOptionException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/internal/Row.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/NonOptionArgumentSpec.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/OptionSpecBuilder.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/MissingRequiredOptionsException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/util/KeyValuePair.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/internal/AbbreviationMap.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/internal/Rows.java
*

/opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/internal/Columns.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/IllegalOptionSpecificationException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/internal/Strings.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/OptionMissingRequiredArgumentException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/OptionSet.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/ArgumentAcceptingOptionSpec.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/util/InetAddressConverter.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/OptionException.java
*
/opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/BuiltinHelpFormatter.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/OptionSpecTokenizer.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/internal/ReflectionException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/internal/OptionNameMap.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/HelpFormatter.java

# 1.93 jackson-annotations 2.12.4

## 1.93.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.94 jackson-datatype-jsr310 2.12.4

## 1.94.1 Available under license :

This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# 1.95 swagger-parser-core 2.0.26

## 1.95.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright 2020 SmartBear Software Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.96 swagger-parser-v3 2.0.26

## 1.96.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2020 SmartBear Software Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.97 jackson-dataformat-yaml 2.12.4

## 1.97.1 Available under license :

This copy of Jackson JSON processor YAML module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

# 1.98 jackson-xc 2.12.4

## 1.98.1 Available under license :

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.
This copy of Jackson JSON processor `jackson-module-jaxb-annotations` module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# 1.99 istack-commons-runtime 3.0.8

## 1.99.1 Available under license :

Copyright (c) 2017 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

- Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Notices for Eclipse Implementation of JAXB

This content is produced and maintained by the Eclipse Implementation of JAXB project.

* Project home: https://projects.eclipse.org/projects/ee4j.jaxb-impl

## Trademarks

Eclipse Implementation of JAXB is a trademark of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Distribution License v. 1.0 which is available at http://www.eclipse.org/org/documents/edl-v10.php.

SPDX-License-Identifier: BSD-3-Clause

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jaxb-ri
* https://github.com/eclipse-ee4j/jaxb-istack-commons
* https://github.com/eclipse-ee4j/jaxb-dtd-parser

* https://github.com/eclipse-ee4j/jaxb-fi
*
 https://github.com/eclipse-ee4j/jaxb-stax-ex
* https://github.com/eclipse-ee4j/jax-rpc-ri

## Third-party Content

This project leverages the following third party content.

Apache Ant (1.10.2)

* License: Apache-2.0 AND W3C AND LicenseRef-Public-Domain

Apache Ant (1.10.2)

* License: Apache-2.0 AND W3C AND LicenseRef-Public-Domain

Apache Felix (1.2.0)

* License: Apache License, 2.0

args4j (2.33)

* License: MIT License

dom4j (1.6.1)

* License: Custom license based on Apache 1.1

file-management (3.0.0)

* License: Apache-2.0
* Project: https://maven.apache.org/shared/file-management/
* Source:
  https://svn.apache.org/viewvc/maven/shared/tags/file-management-3.0.0/

JUnit (4.12)

* License: Eclipse Public License

JUnit (4.12)

* License: Eclipse Public License

maven-compat (3.5.2)

* License: Apache-2.0
* Project: https://maven.apache.org/ref/3.5.2/maven-compat/

* Source:
  https://mvnrepository.com/artifact/org.apache.maven/maven-compat/3.5.2

maven-core
 (3.5.2)

* License: Apache-2.0
* Project: https://maven.apache.org/ref/3.5.2/maven-core/index.html
* Source: https://mvnrepository.com/artifact/org.apache.maven/maven-core/3.5.2

maven-plugin-annotations (3.5)

* License: Apache-2.0
* Project: https://maven.apache.org/plugin-tools/maven-plugin-annotations/
* Source:
  https://github.com/apache/maven-plugin-tools/tree/master/maven-plugin-annotations

maven-plugin-api (3.5.2)

* License: Apache-2.0

maven-resolver-api (1.1.1)

* License: Apache-2.0

maven-resolver-api (1.1.1)

* License: Apache-2.0

maven-resolver-connector-basic (1.1.1)

* License: Apache-2.0

maven-resolver-impl (1.1.1)

* License: Apache-2.0

maven-resolver-spi (1.1.1)

* License: Apache-2.0

maven-resolver-transport-file (1.1.1)

* License: Apache-2.0
* Project: https://maven.apache.org/resolver/maven-resolver-transport-file/
* Source:
  https://github.com/apache/maven-resolver/tree/master/maven-resolver-transport-file

maven-resolver-util (1.1.1)

* License: Apache-2.0

maven-settings
(3.5.2)

* License: Apache-2.0
* Source:
https://mvnrepository.com/artifact/org.apache.maven/maven-settings/3.5.2

OSGi Service Platform Core Companion Code (6.0)

* License: Apache License, 2.0

plexus-archiver (3.5)

* License: Apache-2.0
* Project: https://codehaus-plexus.github.io/plexus-archiver/
* Source: https://github.com/codehaus-plexus/plexus-archiver

plexus-io (3.0.0)

* License: Apache-2.0

plexus-utils (3.1.0)

* License: Apache- 2.0 or Apache- 1.1 or BSD or Public Domain or Indiana
University Extreme! Lab Software License V1.1.1 (Apache 1.1 style)

relaxng-datatype (1.0)

* License: New BSD license

Sax (0.2)

* License: SAX-PD
* Project: http://www.megginson.com/downloads/SAX/
* Source: http://sourceforge.net/project/showfiles.php?group_id=29449

testng (6.14.2)

* License: Apache-2.0 AND (MIT OR GPL-1.0+)
* Project: https://testng.org/doc/index.html
* Source: https://github.com/cbeust/testng

wagon-http-lightweight (3.0.0)

* License: Pending
*
 Project: https://maven.apache.org/wagon/
* Source:
  https://mvnrepository.com/artifact/org.apache.maven.wagon/wagon-http-lightweight/3.0.0

xz for java (1.8)

* License: LicenseRef-Public-Domain

## Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

# 1.100 asm-util 9.1

## 1.100.1 Available under license :

No license file was found, but licenses were detected in source scan.

// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
//    this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/TraceMethodVisitor.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/TraceModuleVisitor.java
*
 /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/CheckClassAdapter.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/Textifier.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-

jar/org/objectweb/asm/util/TraceAnnotationVisitor.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/CheckModuleAdapter.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/CheckSignatureAdapter.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/CheckFieldAdapter.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/TraceRecordComponentVisitor.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/TraceSignatureVisitor.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/ASMifier.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/Printer.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/CheckAnnotationAdapter.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/TraceFieldVisitor.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/CheckRecordComponentAdapter.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/TraceClassVisitor.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/CheckMethodAdapter.java
No license file was found, but licenses were detected in source scan.


<!--
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
-->

Found in path(s):
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/package.html
No license file was found, but licenses were detected in source scan.

/**
* ASM: a very small and fast Java bytecode manipulation framework Copyright (c) 2000-2011 INRIA,
* France Telecom All rights reserved.
*
* <p>Redistribution and use in source and binary forms, with or without modification, are permitted
* provided that the following conditions are met: 1. Redistributions of source code must retain the
* above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions
* in binary form must reproduce the above copyright notice, this list of conditions and the
* following disclaimer in the documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its contributors may be used to
* endorse or promote products derived from this software without specific prior written permission.
*
* <p>THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY
 EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
* FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
* WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY
* WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-

jar/org/objectweb/asm/util/TextifierSupport.java
* /opt/cola/permits/1189734828_1721147053.100699/0/asm-util-9-1-sources-jar/org/objectweb/asm/util/ASMifierSupport.java

# 1.101 asm-tree 9.1

## 1.101.1 Available under license :

No license file was found, but licenses were detected in source scan.


<!--
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
-->


Found in path(s):
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/package.html
No license file was found, but licenses were detected in source scan.


// All rights reserved.
// Redistribution and use in source and binary forms, with or without

// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
//    this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/TypeInsnNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/LabelNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/MethodNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/TryCatchBlockNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/TableSwitchInsnNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/ModuleNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/LookupSwitchInsnNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/FieldInsnNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/FieldNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/ModuleProvideNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/LocalVariableNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/IincInsnNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/InsnNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/LdcInsnNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/MethodInsnNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/ParameterNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/ModuleRequireNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-

jar/org/objectweb/asm/tree/LocalVariableAnnotationNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/InvokeDynamicInsnNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/IntInsnNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/MultiANewArrayInsnNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/RecordComponentNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/InnerClassNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/FrameNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/TypeAnnotationNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/AnnotationNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/ModuleExportNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/LineNumberNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/VarInsnNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/Util.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/ClassNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/InsnList.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/JumpInsnNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/ModuleOpenNode.java
* /opt/cola/permits/1189735964_1701919409.4560196/0/asm-tree-9-1-sources-jar/org/objectweb/asm/tree/AbstractInsnNode.java

# 1.102 asm-commons 9.1

## 1.102.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bundle-DocURL: http://asm.ow2.org
Bundle-License: BSD-3-Clause;link=https://asm.ow2.io/LICENSE.txt
Bundle-ManifestVersion: 2
Bundle-Name: org.objectweb.asm.commons

Bundle-RequiredExecutionEnvironment: J2SE-1.5
Bundle-SymbolicName: org.objectweb.asm.commons
Bundle-Version: 9.1.0
Export-Package: org.objectweb.asm.commons;version="9.1";uses:="org.obj
ectweb.asm,org.objectweb.asm.signature,org.objectweb.asm.tree"
Implementation-Title: Usefull class adapters based on ASM, a very smal
l and fast Java bytecode manipulation framework
Implementation-Version: 9.1
Import-Package: org.objectweb.asm;version="[9.1,10)",org.objectweb.asm
.signature;version="[9.1,10)",org.objectweb.asm.tree;version="[9.1,10
)"
Module-Requires: org.objectweb.asm;transitive=true,org.objectweb.asm.t
ree;transitive=true,org.objectweb.asm.tree.analysis;transitive=true

Found in path(s):
* /opt/cola/permits/1189735679_1628728682.12/0/asm-commons-9-1-1-jar/META-INF/MANIFEST.MF

# 1.103 asm-analysis 9.1

## 1.103.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bundle-DocURL: http://asm.ow2.org
Bundle-License: BSD-3-Clause;link=https://asm.ow2.io/LICENSE.txt
Bundle-ManifestVersion: 2
Bundle-Name: org.objectweb.asm.tree.analysis
Bundle-RequiredExecutionEnvironment: J2SE-1.5
Bundle-SymbolicName: org.objectweb.asm.tree.analysis
Bundle-Version: 9.1.0
Export-Package: org.objectweb.asm.tree.analysis;version="9.1";uses:="o
rg.objectweb.asm,org.objectweb.asm.tree"
Implementation-Title: Static code analysis API of ASM, a very small an
d fast Java bytecode manipulation framework
Implementation-Version: 9.1
Import-Package: org.objectweb.asm;version="[9.1,10)",org.objectweb.asm
.tree;version="[9.1,10)"
Module-Requires: org.objectweb.asm.tree;transitive=true

Found in path(s):
* /opt/cola/permits/1189736546_1701919423.6275084/0/asm-analysis-9-1-jar/META-INF/MANIFEST.MF

# 1.104 protobuf-java-util 3.17.2

## 1.104.1 Available under license :

No license file was found, but licenses were detected in source scan.

// Copyright 2008 Google Inc.  All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
//     * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//     * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
//     * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1193479768_1629362526.67/0/protobuf-java-util-3-17-2-sources-jar/com/google/protobuf/util/Structs.java
* /opt/cola/permits/1193479768_1629362526.67/0/protobuf-java-util-3-17-2-sources-jar/com/google/protobuf/util/Durations.java
* /opt/cola/permits/1193479768_1629362526.67/0/protobuf-java-util-3-17-2-sources-jar/com/google/protobuf/util/Values.java
* /opt/cola/permits/1193479768_1629362526.67/0/protobuf-java-util-3-17-2-sources-jar/com/google/protobuf/util/TimeUtil.java
* /opt/cola/permits/1193479768_1629362526.67/0/protobuf-java-util-3-17-2-sources-jar/com/google/protobuf/util/FieldMaskUtil.java
* /opt/cola/permits/1193479768_1629362526.67/0/protobuf-java-util-3-17-2-sources-jar/com/google/protobuf/util/Timestamps.java
* /opt/cola/permits/1193479768_1629362526.67/0/protobuf-java-util-3-17-2-sources-jar/com/google/protobuf/util/FieldMaskTree.java
* /opt/cola/permits/1193479768_1629362526.67/0/protobuf-java-util-3-17-2-sources-jar/com/google/protobuf/util/JsonFormat.java

# 1.105 protobuf-java 3.17.2

## 1.105.1 Available under license :

No license file was found, but licenses were detected in source scan.

// Copyright 2008 Google Inc.  All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
//     * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//     * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
//     * Neither the name of Google Inc. nor the names of its

// this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/NioByteString.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/UnknownFieldSetLiteSchema.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/TextFormat.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ProtobufArrayList.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/IterableByteBufferInputStream.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MapFieldSchemaLite.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ExtensionSchemaLite.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ProtocolMessageEnum.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/google/protobuf/wrappers.proto
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/CodedInputStream.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/google/protobuf/duration.proto
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/google/protobuf/descriptor.proto
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/RepeatedFieldBuilder.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MessageLiteToString.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/CodedInputStreamReader.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/SchemaUtil.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/NewInstanceSchemas.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MapEntryLite.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ManifestSchemaFactory.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/SmallSortedMap.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/DescriptorMessageInfoFactory.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MessageSchema.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-

jar/com/google/protobuf/ServiceException.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MessageLite.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/BlockingRpcChannel.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/LazyFieldLite.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MapField.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/UnknownFieldSetLite.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/UnknownFieldSetSchema.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ExperimentalApi.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/InvalidProtocolBufferException.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/Extension.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MapFieldSchemaFull.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/LazyField.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/PrimitiveNonBoxingCollection.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/google/protobuf/struct.proto
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/BlockingService.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/BinaryWriter.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ExtensionSchemaFull.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/UnmodifiableLazyStringList.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/BinaryReader.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/OneofInfo.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/AbstractProtobufList.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/GeneratedMessage.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MapFieldLite.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-

jar/com/google/protobuf/IntArrayList.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/FieldInfo.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/CodedOutputStream.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/FloatArrayList.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/DoubleArrayList.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/RepeatedFieldBuilderV3.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ByteBufferWriter.java
*
/opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/LongArrayList.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/GeneratedMessageInfoFactory.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/NewInstanceSchema.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ExtensionLite.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/SingleFieldBuilder.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/GeneratedMessageLite.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MessageLiteOrBuilder.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/SchemaFactory.java
*
/opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/StructuralMessageInfo.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/google/protobuf/any.proto
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/DynamicMessage.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MessageReflection.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/GeneratedMessageV3.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/RpcCallback.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ExtensionSchemas.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/Reader.java
*

/opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/NewInstanceSchemaFull.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/FieldType.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MutabilityOracle.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/BooleanArrayList.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/AbstractMessageLite.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/UninitializedMessageException.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MapEntry.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ByteOutput.java
*
/opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/TypeRegistry.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/Writer.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/Message.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/AbstractParser.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/LazyStringList.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MapFieldSchema.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/UnsafeByteOperations.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/JavaType.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/Protobuf.java
*
/opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/RpcChannel.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ExtensionSchema.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/CodedOutputStreamWriter.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/UnknownFieldSchema.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/Android.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/RpcUtil.java

* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ArrayDecoders.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/Schema.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/Internal.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/Descriptors.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MessageSetSchema.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/LazyStringArrayList.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/UnknownFieldSet.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/NewInstanceSchemaLite.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/TextFormatParseInfoTree.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/google/protobuf/empty.proto
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/google/protobuf/timestamp.proto
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ExtensionRegistryLite.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ProtocolStringList.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ListFieldSchema.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/google/protobuf/api.proto
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/DiscardUnknownFieldsParser.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/RopeByteString.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/google/protobuf/type.proto
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ByteString.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/TextFormatParseLocation.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MessageInfo.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/google/protobuf/compiler/plugin.proto
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/Service.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ProtobufLists.java

* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MapFieldSchemas.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/google/protobuf/source_context.proto
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MessageInfoFactory.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/SingleFieldBuilderV3.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/google/protobuf/field_mask.proto
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/UnsafeUtil.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/TextFormatEscaper.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/FieldSet.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/RawMessageInfo.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ProtoSyntax.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/MessageOrBuilder.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/WireFormat.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ExtensionRegistry.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/AllocatedBuffer.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/Parser.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/AbstractMessage.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/Utf8.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/BufferAllocator.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/ExtensionRegistryFactory.java
* /opt/cola/permits/1193480003_1629362448.03/0/protobuf-java-3-17-2-sources-jar/com/google/protobuf/RpcController.java

# 1.106 error_prone_annotations 2.4.0

## 1.106.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/concurrent/UnlockMethod.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/concurrent/LockMethod.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/NoAllocation.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2016 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/FormatMethod.java

* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/MustBeClosed.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/RestrictedApi.java
*
 /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/FormatString.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/CompatibleWith.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/DoNotMock.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2017 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/concurrent/GuardedBy.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/DoNotCall.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/OverridingMethodsMustInvokeSuper.java
*
 /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/CheckReturnValue.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2015 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
```

*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/RequiredModifiers.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/CompileTimeConstant.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/Var.java
*
 /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/CanIgnoreReturnValue.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/Immutable.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/concurrent/LazyInit.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/ForOverride.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/SuppressPackageLocation.java
* /opt/cola/permits/1199767181_1630583089.73/0/error-prone-annotations-2-4-0-sources-2-jar/com/google/errorprone/annotations/IncompatibleModifiers.java

# 1.107 animal-sniffer-annotation 1.19

## 1.107.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* The MIT License
*
* Copyright (c) 2008 Kohsuke Kawaguchi and codehaus.org.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
* copies of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
*

* The above copyright notice and this permission notice shall be included in
* all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
* AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
* THE SOFTWARE.
*
*/

Found in path(s):
* /opt/cola/permits/1199767196_1630583071.26/0/animal-sniffer-annotations-1-19-sources-1-jar/org/codehaus/mojo/animal_sniffer/IgnoreJRERequirement.java
No license file was found, but licenses were detected in source scan.

The MIT License

Copyright (c) 2009 codehaus.org.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Found in path(s):
* /opt/cola/permits/1199767196_1630583071.26/0/animal-sniffer-annotations-1-19-sources-1-jar/META-INF/maven/org.codehaus.mojo/animal-sniffer-annotations/pom.xml

# 1.108 error_prone_annotations 2.7.1

## 1.108.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2016 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/CompatibleWith.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/DoNotMock.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/FormatMethod.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/MustBeClosed.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/RestrictedApi.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/FormatString.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2017 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
```

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/DoNotCall.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/OverridingMethodsMustInvokeSuper.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/CheckReturnValue.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/concurrent/GuardedBy.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/CompileTimeConstant.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/Immutable.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/ForOverride.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/SuppressPackageLocation.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/IncompatibleModifiers.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/CanIgnoreReturnValue.java

* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/concurrent/LazyInit.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/RequiredModifiers.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/Var.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/concurrent/UnlockMethod.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/concurrent/LockMethod.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/NoAllocation.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2021 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
```

*/

Found in path(s):
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/InlineMe.java
* /opt/cola/permits/1201423115_1631149789.85/0/error-prone-annotations-2-7-1-sources-1-jar/com/google/errorprone/annotations/InlineMeValidationDisabled.java

# 1.109 jackson-datatype-guava 2.12.4

## 1.109.1 Available under license :

This copy of Jackson JSON processor `jackson-datatype-guava` module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# 1.110 jackson-datatype-joda 2.12.4

## 1.110.1 Available under license :

This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# 1.111 bcprov-jdk15on 1.67

## 1.111.1 Available under license :

No license file was found, but licenses were detected in source scan.

/**
* SPHINCS-256 signer.
* <p>
* This implementation is heavily based on the reference implementation in SUPERCOP, the main difference being the digests used
* for message hashing and tree construction are now configurable (within limits...) and that the implementation produces
* detached signatures.

```
* </p>
* <p>
* The SPHINCS reference implementation is public domain, as per the statement in the second last paragraph of
* section 1 in https://eprint.iacr.org/2014/795.pdf
* </p>
*/
```

Found in path(s):
* /opt/cola/permits/1205937200_1632324303.64/0/bcprov-jdk15on-1-67-sources-jar/org/bouncycastle/pqc/crypto/sphincs/SPHINCS256Signer.java
No license file was found, but licenses were detected in source scan.

```
/**
   * Magic value for proprietary OpenSSH private key.
   **/
```

Found in path(s):
* /opt/cola/permits/1205937200_1632324303.64/0/bcprov-jdk15on-1-67-sources-jar/org/bouncycastle/crypto/util/OpenSSHPrivateKeyUtil.java
No license file was found, but licenses were detected in source scan.

```
/**
* The Bouncy Castle License
*
* Copyright (c) 2000-2019 The Legion Of The Bouncy Castle Inc. (http://www.bouncycastle.org)
* <p>
* Permission is hereby granted, free of charge, to any person obtaining a copy of this software
* and associated documentation files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense,
* and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
* subject to the following conditions:
* <p>
* The above copyright notice and this permission notice shall be included in all copies or substantial
* portions of the Software.
* <p>
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
* INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
* PURPOSE AND NONINFRINGEMENT.
 IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
* OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
* DEALINGS IN THE SOFTWARE.
*/
```

Found in path(s):
* /opt/cola/permits/1205937200_1632324303.64/0/bcprov-jdk15on-1-67-sources-jar/org/bouncycastle/LICENSE.java
No license file was found, but licenses were detected in source scan.

public DomainParameters(BigInteger p, BigInteger g, BigInteger q, BigInteger j,

Found in path(s):
* /opt/cola/permits/1205937200_1632324303.64/0/bcprov-jdk15on-1-67-sources-jar/org/bouncycastle/asn1/x9/DomainParameters.java
No license file was found, but licenses were detected in source scan.

/**
* Poly1305 message authentication code, designed by D. J. Bernstein.
* <p>
* Poly1305 computes a 128-bit (16 bytes) authenticator, using a 128 bit nonce and a 256 bit key
* consisting of a 128 bit key applied to an underlying cipher, and a 128 bit key (with 106
* effective key bits) used in the authenticator.
* <p>
* The polynomial calculation in this implementation is adapted from the public domain <a
* href="https://github.com/floodyberry/poly1305-donna">poly1305-donna-unrolled</a> C implementation
* by Andrew M (@floodyberry).
* @see Poly1305KeyGenerator
*/

Found in path(s):
* /opt/cola/permits/1205937200_1632324303.64/0/bcprov-jdk15on-1-67-sources-jar/org/bouncycastle/crypto/macs/Poly1305.java
No license file was found, but licenses were detected in source scan.

// now keep space is free to use

Found in path(s):
* /opt/cola/permits/1205937200_1632324303.64/0/bcprov-jdk15on-1-67-sources-jar/org/bouncycastle/pqc/crypto/gmss/GMSSPrivateKeyParameters.java
No license file was found, but licenses were detected in source scan.

/**
* Copyright (c) 2011 Tim Buktu (tbuktu@hotmail.com)
*
* Permission is hereby granted, free of charge, to any person obtaining a
* copy of this software and associated documentation files (the "Software"),
* to deal in the Software without restriction, including without limitation
* the rights to use, copy, modify, merge, publish, distribute, sublicense,
* and/or sell copies of the Software, and to permit persons to whom the
* Software is furnished to do so, subject to the following conditions:
*
* The above copyright notice and this permission notice shall be included in

* all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
* AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
 CLAIM, DAMAGES OR OTHER
* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
* FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
* DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/1205937200_1632324303.64/0/bcprov-jdk15on-1-67-sources-jar/org/bouncycastle/pqc/math/ntru/euclid/test/IntEuclideanTest.java

# 1.112 bcpkix-jdk15on 1.67

## 1.112.1 Available under license :

No license file was found, but licenses were detected in source scan.

/**
 * Base constructor.
 * 
 * The authorizer attempts to perform matching (including the use of the wildcard) in accordance with RFC 6125.
 * 
 * 
 * Known suffixes is a list of public domain suffixes that can't be used as wild cards for
 * example *.com, or c*c.com, as a dns wildcard could match every/most .com domains if a registrar were issue it.
 * If *.com is in the known suffixes list will not be allowed to match.
 * 
 *
 * @param knownSuffixes a set of suffixes that cannot be wild-carded, e.g. { ".com", ".net", ".org" }
 */

Found in path(s):
* /opt/cola/permits/1205937193_1632324182.11/0/bcpkix-jdk15on-1-67-sources-jar/org/bouncycastle/est/jcajce/JsseDefaultHostnameAuthorizer.java

# 1.113 guava 31.0.1-jre

## 1.113.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Guava Authors

*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/cache/LongAddable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/escape/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/cache/LongAddables.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/html/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/LongAddables.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/RateLimiter.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/ChecksumHashFunction.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/base/StandardSystemProperty.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/AbstractByteHasher.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/io/ByteSink.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/io/CharSink.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/reflect/ClassPath.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/reflect/ImmutableTypeToInstanceMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/reflect/AbstractInvocationHandler.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/io/ByteSource.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/io/FileWriteMode.java

* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/CharSource.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/CartesianList.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ListenableScheduledFuture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableRangeMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/FilteredKeyMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/BaseEncoding.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/reflect/TypeToInstanceMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/math/PairedStats.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/math/Stats.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/xml/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/Closer.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/reflect/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/SmoothRateLimiter.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableRangeSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/reflect/MutableTypeToInstanceMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/reflect/Invokable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/SipHashFunction.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/reflect/Parameter.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ServiceManager.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/math/PairedStatsAccumulator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/LongAddable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/reflect/TypeCapture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-

jar/com/google/common/math/LinearTransformation.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/math/StatsAccumulator.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/JdkBackedImmutableMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/ImmutableSupplier.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/ExecutionSequencer.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/JdkBackedImmutableSet.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Doubles}. Intended to be empty for regular

* version.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/DoublesMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Outer class that exists solely to let us write {@code Partially.GwtIncompatible} instead of plain
* {@code GwtIncompatible}. This is more accurate for {@link Futures#catching}, which is available
* under GWT but with a slightly different signature.
*
* <p>We can't use {@code PartiallyGwtIncompatible} because then the GWT compiler wouldn't recognize
* it as a {@code GwtIncompatible} annotation. And for {@code Futures.catching}, we need the GWT
* compiler to autostrip the normal server method in order to expose the special, inherited GWT
* version.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/Partially.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/reflect/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/math/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/xml/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/eventbus/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ParametricNullness.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/ParametricNullness.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/ParametricNullness.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/net/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/ParametricNullness.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/reflect/ParametricNullness.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/escape/ParametricNullness.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/ParametricNullness.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/html/ParametricNullness.java
*

/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/xml/ParametricNullness.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/math/ParametricNullness.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/ParametricNullness.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/net/ParametricNullness.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/cache/ParametricNullness.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/eventbus/ParametricNullness.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ElementTypesAreNonnullByDefault.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/html/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/escape/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/cache/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ParametricNullness.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/Traverser.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/BaseGraph.java

* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ClosingFuture.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/AbstractBaseGraph.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/IndexedImmutableSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/BaseImmutableMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableMap.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableBiMap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/IncidentEdgeSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/CompactHashing.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-

jar/com/google/common/collect/ClassToInstanceMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/LexicographicalOrdering.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/MutableClassToInstanceMap.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Iterators.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractListMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingConcurrentMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ReverseNaturalOrdering.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableList.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/UsingToStringOrdering.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ByFunctionOrdering.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/BiMap.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Multimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractMapBasedMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Interner.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/SingletonImmutableSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/HashMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingSortedMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/NullsLastOrdering.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingSet.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingList.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/SetMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ArrayListMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingCollection.java

* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractIterator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingIterator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/MapDifference.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingSortedSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/NaturalOrdering.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Maps.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractSetMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingQueue.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Multimaps.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/LinkedHashMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/CompoundOrdering.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/SortedSetMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingObject.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/LinkedHashMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Synchronized.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingListIterator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Multisets.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-

jar/com/google/common/collect/ComparatorOrdering.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ListMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Iterables.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractMapEntry.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ReverseOrdering.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/TreeMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Ordering.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/NullsFirstOrdering.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/EnumBiMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingMapEntry.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Sets.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractSortedSetMultimap.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/EnumHashBiMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Multiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Lists.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/TreeMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/LinkedListMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/HashMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ConcurrentHashMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractBiMap.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ExplicitOrdering.java

* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractMapBasedMultiset.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/math/BigDecimalMath.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/OverflowAvoidingLockSupport.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/Java8Compatibility.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/Java8Compatibility.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/math/ToDoubleRounder.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Java8Compatibility.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.

```
*/
/**
* Holder for web specializations of methods of {@code Ints}. Intended to be empty for regular
* version.
*/
```

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/IntsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/PatternCompiler.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/JdkPattern.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/CommonPattern.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/CommonMatcher.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
  * Not supported. <b>You are attempting to create a map that may contain a non-{@code Comparable}
  * key.</b> Proper calls will resolve to the version in {@code ImmutableSortedMap}, not this dummy
  * version.
  *
  * @throws UnsupportedOperationException always
  * @deprecated <b>Pass a key of type {@code Comparable}
 to use {@link
  *     ImmutableSortedMap#of(Comparable, Object)}.</b>
  */
```

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedMapFauxverideShim.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
  * This method was written by Doug Lea with assistance from members of JCP JSR-166 Expert Group
  * and released to the public domain, as explained at
  * http://creativecommons.org/licenses/publicdomain
  *
  * As of 2010/06/11, this method is identical to the (package private) hash method in OpenJDK 7's
  * java.util.HashMap
 class.
  */
```

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/Striped.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableBiMapFauxverideShim.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/CollectSpliterators.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
  * Returns an array containing all of the elements in the specified collection. This method
  * returns the elements in the order they are returned by the collection's iterator. The returned
  * array is "safe" in that no references to it are maintained by the collection. The caller is
  * thus free to modify the returned
```

array.
 *
 * <p>This method assumes that the collection size doesn't change while the method is running.
 *
 * <p>TODO(kevinb): support concurrently modified collections?
 *
 * @param c the collection for which to return an array of elements
 */

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ObjectArrays.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby disclaims
copyright to this source code.
 */

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/SmallCharMatcher.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Shorts}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/ShortsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/GwtTransient.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/AtomicLongMap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for extra methods of {@code Objects} only in web. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/ExtraObjectsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/MacHashFunction.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ConsumingQueueIterator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/AsyncCallable.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/LittleEndianByteArray.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/ReaderInputStream.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/AggregateFutureState.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-

jar/com/google/common/hash/FarmHashFingerprint64.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/CombinedFuture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/InterruptibleTask.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/Platform.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/ImmutableMapEntry.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/io/MoreFiles.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/MultimapBuilder.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/reflect/Reflection.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableMapEntrySet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/StandardTable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Table.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Collections2.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/TreeBasedTable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/HashBasedTable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableListMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableMapKeySet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Platform.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Range.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableMapValues.java
*

/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/EmptyImmutableListMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Tables.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/PeekingIterator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Serialization.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableEntry.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableBiMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/StandardRowSortedTable.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/CollectPreconditions.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableBiMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/UnmodifiableIterator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/SingletonImmutableBiMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/thirdparty/publicsuffix/PublicSuffixPatterns.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableCollection.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/AbstractHasher.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/HashFunction.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/FutureCallback.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/cache/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/cache/RemovalNotification.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/AsyncFunction.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/cache/CacheBuilderSpec.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/Funnels.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/Hashing.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/EmptyContiguousSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Optional.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/DescendingImmutableSortedMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/HashCode.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/UnsignedInts.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/Uninterruptibles.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingExecutorService.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/TreeRangeSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/cache/ForwardingLoadingCache.java

*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/base/Absent.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/AbstractCompositeHashFunction.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/CycleDetectingLockFactory.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/WrappingExecutorService.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/Queues.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingListeningExecutorService.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/BloomFilter.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/PrimitiveSink.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/reflect/Types.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/base/Ticker.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/Hasher.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/MessageDigestHashFunction.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/UncheckedExecutionException.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/math/BigIntegerMath.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/base/Enums.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/cache/RemovalListeners.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/net/MediaType.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/base/FunctionalEquivalence.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/cache/CacheLoader.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/cache/RemovalCause.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/net/HostAndPort.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractListeningExecutorService.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-

jar/com/google/common/cache/Weigher.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/cache/ForwardingCache.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/primitives/UnsignedLong.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/cache/CacheStats.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/primitives/UnsignedLongs.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/math/LongMath.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/cache/LoadingCache.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/cache/Cache.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/math/DoubleMath.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/net/HttpHeaders.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/AbstractStreamingHasher.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/AbstractNonStreamingHashFunction.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/base/PairwiseEquivalence.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/ListeningScheduledExecutorService.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/reflect/TypeParameter.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/base/Present.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/Crc32cHashFunction.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/cache/RemovalListener.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractScheduledService.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/cache/AbstractCache.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/RegularContiguousSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/BloomFilterStrategies.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/math/MathPreconditions.java

* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractSortedMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/UnsignedInteger.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/Murmur3_32HashFunction.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/Murmur3_128HashFunction.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/Funnel.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/math/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/math/IntMath.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/math/DoubleUtils.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/ParseRequest.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/cache/AbstractLoadingCache.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/HashingOutputStream.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ExecutionError.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/BoundType.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/*
  * This method was rewritten in Java from an intermediate step of the Murmur hash function in
  * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
```

```
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby
disclaims copyright to this source code.
 */
```

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Hashing.java
No license file was found, but licenses were detected in source scan.

```
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*/
```

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/cache/Striped64.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/cache/LongAdder.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/AtomicDoubleArray.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/Striped64.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/LongAdder.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-

jar/com/google/common/collect/ForwardingSortedMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/GeneralRange.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSortedMultisetFauxverideShim.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/Count.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/RangeSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/SortedIterables.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/AbstractRangeSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSortedMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableSortedMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/SortedIterable.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not
* use this file except in compliance with the License. You may obtain a copy of
* the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/SortedMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/SortedMultisets.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 The Guava Authors
*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/base/VerifyException.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/base/Verify.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/FilteredMultimapValues.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/io/CharSequenceReader.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/AbstractTable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/reflect/TypeVisitor.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/eventbus/SubscriberExceptionHandler.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/eventbus/SubscriberExceptionContext.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/WrappingScheduledExecutorService.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/hash/HashingInputStream.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/base/Utf8.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/Runnables.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingListMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingSetMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/UnmodifiableListIterator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/MinMaxPriorityQueue.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractSequentialIterator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/RowSortedTable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingImmutableCollection.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/SortedMapDifference.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingSortedSetMultimap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/StandardMutableNetwork.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/StandardMutableValueGraph.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/ForwardingValueGraph.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/GraphConstants.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/MutableValueGraph.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/UndirectedMultiNetworkConnections.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Comparators.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/AbstractValueGraph.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/GraphBuilder.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ArrayListMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/DirectedNetworkConnections.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/AbstractUndirectedNetworkConnections.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/RangeGwtSerializationDependencies.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/DirectedMultiNetworkConnections.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/StandardNetwork.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/CollectCollectors.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableMultisetGwtSerializationDependencies.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/MoreCollectors.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/AbstractGraphBuilder.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/MapIteratorCache.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/EdgesConnecting.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-

jar/com/google/common/graph/EndpointPair.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/ForwardingGraph.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/EndpointPairIterator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/AbstractGraph.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/DirectedGraphConnections.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/StandardValueGraph.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/ElementOrder.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/NetworkConnections.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/LinkedHashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/UndirectedNetworkConnections.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/MultiEdgesConnecting.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/StandardMutableGraph.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/GraphConnections.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/MapRetrievalCache.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/ForwardingNetwork.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/AbstractDirectedNetworkConnections.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/ImmutableValueGraph.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/NetworkBuilder.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/AbstractNetwork.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/ValueGraph.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/HashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/ValueGraphBuilder.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/graph/UndirectedGraphConnections.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/LittleEndianDataInputStream.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/eventbus/AllowConcurrentEvents.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/FinalizableReference.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/eventbus/Subscribe.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/CharStreams.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/AbstractIterator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Functions.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Defaults.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/ByteStreams.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Supplier.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/HashBiMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/Resources.java

* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/EnumMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/eventbus/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/Flushables.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/Files.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Throwables.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/eventbus/AsyncEventBus.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Interners.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Suppliers.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Predicates.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Charsets.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/eventbus/EventBus.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/FinalizablePhantomReference.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/eventbus/DeadEvent.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/LittleEndianDataOutputStream.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractFuture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Function.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/LineBuffer.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/CountingOutputStream.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ExecutionList.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/FinalizableSoftReference.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Preconditions.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ListenableFuture.java
*

/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/Primitives.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Predicate.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/LineReader.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/FinalizableReferenceQueue.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Objects.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/DirectExecutor.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/FinalizableWeakReference.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/CountingInputStream.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/MultiInputStream.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/Closeables.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you
* may not use this file except in compliance with the License.  You may
* obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
* implied.  See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Streams.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableEnumSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/DiscreteDomain.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableClassToInstanceMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ComparisonChain.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableAsList.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/SingletonImmutableList.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ArrayTable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/SingletonImmutableTable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedSetFauxverideShim.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractIndexedListIterator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/TableCollectors.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableSortedSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableTable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableList.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-

jar/com/google/common/collect/EmptyImmutableSetMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/ForwardingTable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/ComputationException.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/ImmutableSetMultimap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/ServiceManagerBridge.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-

jar/com/google/common/primitives/Platform.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/Internal.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/FilteredKeySetMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/TreeTraverser.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/AbstractSortedKeySortedSetMultimap.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/AllEqualOrdering.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/RegularImmutableAsList.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingBlockingDeque.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/UnmodifiableSortedMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/SortedMultisetBridge.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/ForwardingImmutableMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/RangeMap.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/ImmutableEnumMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/collect/FilteredKeyListMultimap.java

* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/FilteredEntryMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/TransformedListIterator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/TreeRangeMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/DescendingMultiset.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/CompactHashMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/FilteredSetMultimap.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingNavigableMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingImmutableSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/CompactLinkedHashSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingImmutableList.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingBlockingDeque.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/TransformedIterator.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/DescendingImmutableSortedSet.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingNavigableSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/CompactLinkedHashMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/CompactHashSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ForwardingDeque.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/EvictingQueue.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/FilteredEntrySetMultimap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/AbstractNavigableMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/FilteredMultimap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
  * This following method is a modified version of one found in
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30
  * which contained the following notice:
  *
  * Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to
  *
 the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/
  *
  * Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.
  */

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/MoreExecutors.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-

jar/com/google/common/util/concurrent/ListenableFutureTask.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/internal/Finalizer.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/escape/Escaper.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Converter.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/FileBackedOutputStream.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/thirdparty/publicsuffix/TrieParser.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/FluentIterable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/CharMatcher.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/net/InetAddresses.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Stopwatch.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/Floats.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Joiner.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/escape/UnicodeEscaper.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/Shorts.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/MultiReader.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/Doubles.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/Chars.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/net/PercentEscaper.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/Bytes.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/SequentialExecutor.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/Longs.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/Ints.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/Booleans.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/TrustedListenableFutureTask.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/eventbus/Subscriber.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/MoreObjects.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/math/Quantiles.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ListenerCallQueue.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/eventbus/Dispatcher.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/eventbus/SubscriberRegistry.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
```

*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/ImmutableLongArray.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/hash/AbstractHashFunction.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingCondition.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/ImmutableIntArray.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/ImmutableDoubleArray.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingLock.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/annotations/VisibleForTesting.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/reflect/TypeToken.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/TimeLimiter.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractCatchingFuture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/escape/CharEscaper.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-

jar/com/google/common/util/concurrent/FakeTimeLimiter.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/TimeoutFuture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractTransformFuture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/SimpleTimeLimiter.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/FluentFuture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/io/AppendableWriter.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/AggregateFuture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/io/PatternFilenameFilter.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/ImmediateFuture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/escape/CharEscaperBuilder.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/base/CaseFormat.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/CollectionFuture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/FuturesGetChecked.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-
jar/com/google/common/util/concurrent/GwtFluentFutureCatchingSpecialization.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingListenableFuture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/SignedBytes.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/Service.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Splitter.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/LineProcessor.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/escape/ArrayBasedEscaperMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/cache/ReferenceEntry.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractIdleService.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractService.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/DenseImmutableTable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/ByteProcessor.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/ByteArrayDataOutput.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/reflect/TypeResolver.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/html/HtmlEscapers.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Platform.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/MapMaker.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractExecutionThreadService.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/SettableFuture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/xml/XmlEscapers.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/net/InternetDomainName.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/Cut.java

* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/escape/ArrayBasedUnicodeEscaper.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/ByteArrayDataInput.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/Callables.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/annotations/GwtIncompatible.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/escape/Escapers.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableTable.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/MapMakerInternalMap.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/SparseImmutableTable.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/annotations/GwtCompatible.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/net/HostSpecifier.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/cache/LocalCache.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/JdkFutureAdapters.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/escape/ArrayBasedCharEscaper.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingFluentFuture.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/cache/CacheBuilder.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/escape/Platform.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingFuture.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/net/UrlEscapers.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/UnsignedBytes.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/annotations/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ContiguousSet.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/annotations/Beta.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/Monitor.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/net/package-info.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingBlockingQueue.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ThreadFactoryBuilder.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/SortedLists.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Strings.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Equivalence.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/UncaughtExceptionHandlers.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/Ascii.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/ListeningExecutorService.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/Atomics.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedAsList.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Floats}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/primitives/FloatsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/base/NullnessCasts.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/NullnessCasts.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/util/concurrent/NullnessCasts.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/ImmutableGraph.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/RecursiveDeleteOption.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/SuccessorsFunction.java
*
 /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/PredecessorsFunction.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/Graphs.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/Graph.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/MutableGraph.java

* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/collect/TopKSelector.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/MutableNetwork.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/Network.java
* /opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/graph/ImmutableNetwork.java
*
/opt/cola/permits/1208993118_1632983176.74/0/guava-31-0-1-jre-sources-1-jar/com/google/common/io/InsecureRecursiveDeleteException.java

# 1.114 grpc-netty 1.39.0

## 1.114.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/Http2ControlFrameLimitEncoder.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/CreateStreamCommand.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettyReadableBuffer.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettyServerTransport.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/CancelClientStreamCommand.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettyServerStream.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettyServer.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NegotiationType.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettyClientTransport.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettyChannelBuilder.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettyClientHandler.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/Utils.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/SendGrpcFrameCommand.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettyServerHandler.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettyServerBuilder.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/SendResponseHeadersCommand.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2020 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/InternalNettyServerCredentials.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/InsecureFromHttp1ChannelCredentials.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettyServerCredentials.java
*
 /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettySslContextServerCredentials.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/InternalNettyChannelCredentials.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettySslContextChannelCredentials.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettyChannelCredentials.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/InternalNettyServerBuilder.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/ClientTransportLifecycleManager.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-

jar/io/grpc/netty/GrpcHttp2OutboundHeaders.java
*
/opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-
jar/io/grpc/netty/GracefulCloseCommand.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-
jar/io/grpc/netty/GrpcHttp2ConnectionHandler.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/StreamIdHolder.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-
jar/io/grpc/netty/ForcefulCloseCommand.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-
jar/io/grpc/netty/AbstractHttp2Headers.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-
jar/io/grpc/netty/InternalNettyChannelBuilder.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-
jar/io/grpc/netty/NettyChannelProvider.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/JettyTlsUtil.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-
jar/io/grpc/netty/GrpcSslContexts.java
*
/opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-
jar/io/grpc/netty/ProtocolNegotiator.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-
jar/io/grpc/netty/CancelServerStreamCommand.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-
jar/io/grpc/netty/ProtocolNegotiators.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/WriteQueue.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-
jar/io/grpc/netty/NettyWritableBufferAllocator.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-

jar/io/grpc/netty/NettyClientStream.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettyWritableBuffer.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/SendPingCommand.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettyServerProvider.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/AbstractNettyHandler.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/package-info.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2016 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by
 applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/GrpcHttp2HeadersUtils.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2021 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/FixedTrustManagerFactory.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/FixedKeyManagerFactory.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/UnhelpfulSecurityProvider.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2017 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/KeepAliveEnforcer.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/MaxConnectionIdleManager.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2018 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/InternalNettySocketSupport.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/NettySocketSupport.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2019 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/InternalWriteBufferingAndExceptionHandlerUtils.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/WriteBufferingAndExceptionHandler.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/ProtocolNegotiationEvent.java

*
 /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/InternalProtocolNegotiationEvent.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/InternalProtocolNegotiator.java
* /opt/cola/permits/1219225126_1635198275.53/0/grpc-netty-1-39-0-sources-jar/io/grpc/netty/InternalProtocolNegotiators.java

# 1.115 libcxx 9.0.8svn

## 1.115.1 Available under license :

```
==============================================================================
libc++ License
==============================================================================

The libc++ library is dual licensed under both the University of Illinois
"BSD-Like" license and the MIT license.  As a user of this code you may choose
to use it under either license.  As a contributor, you agree to allow your code
to be used under both.

Full text of the relevant licenses is included below.

==============================================================================

University of Illinois/NCSA
Open Source License

Copyright (c) 2009-2017 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

   LLVM Team

   University of Illinois at Urbana-Champaign

   http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction, including
 without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:
```

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimers in the
  documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at
  Urbana-Champaign, nor the names of its contributors may be used to
  endorse or promote products derived from this Software without specific
  prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==========

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial
portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
# People who have agreed to one of the CLAs and can contribute patches.
# The AUTHORS file lists the copyright holders; this file
# lists people.  For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# Names should be added to this file only after verifying that
# the individual or the individual's organization has agreed to
# the appropriate Contributor License Agreement, found here:
#
# https://developers.google.com/open-source/cla/individual
# https://developers.google.com/open-source/cla/corporate
#
# The agreement for individuals can be filled out on the web.
#
# When adding J Random Contributor's name to this file,
# either J's name or J's organization's name should be
# added to the AUTHORS file, depending on whether the
# individual or corporate CLA was used.
#
# Names should be added to this file as:
#  Name <email address>
#
# Please keep the list sorted.

Albert Pretorius <pretoalb@gmail.com>
Arne Beer <arne@twobeer.de>
Billy
Robert O'Neal III <billy.oneal@gmail.com> <bion@microsoft.com>
Chris Kennelly <ckennelly@google.com> <ckennelly@ckennelly.com>
Christopher Seymour <chris.j.seymour@hotmail.com>
Cyrille Faucheux <cyrille.faucheux@gmail.com>
David Coeurjolly <david.coeurjolly@liris.cnrs.fr>
Deniz Evrenci <denizevrenci@gmail.com>
Dominic Hamon <dma@stripysock.com> <dominic@google.com>

Dominik Czarnota <dominik.b.czarnota@gmail.com>
Eric Fiselier <eric@efcs.ca>
Eugene Zhuk <eugene.zhuk@gmail.com>
Evgeny Safronov <division494@gmail.com>
Federico Ficarelli <federico.ficarelli@gmail.com>
Felix Homann <linuxaudio@showlabor.de>
Ismael Jimenez Martinez <ismael.jimenez.martinez@gmail.com>
Jern-Kuan Leong <jernkuan@gmail.com>
JianXiong Zhou <zhoujianxiong2@gmail.com>
Joao Paulo Magalhaes <joaoppmagalhaes@gmail.com>
John Millikin <jmillikin@stripe.com>
Jussi Knuuttila <jussi.knuuttila@gmail.com>
Kai Wolf <kai.wolf@gmail.com>
Kishan Kumar <kumar.kishan@outlook.com>
Kaito Udagawa <umireon@gmail.com>
Lei Xu <eddyxu@gmail.com>
Matt
Clarkson <mattyclarkson@gmail.com>
Maxim Vafin <maxvafin@gmail.com>
Nick Hutchinson <nshutchinson@gmail.com>
Oleksandr Sochka <sasha.sochka@gmail.com>
Ori Livneh <ori.livneh@gmail.com>
Pascal Leroy <phl@google.com>
Paul Redmond <paul.redmond@gmail.com>
Pierre Phaneuf <pphaneuf@google.com>
Radoslav Yovchev <radoslav.tm@gmail.com>
Raul Marin <rmrodriguez@cartodb.com>
Ray Glover <ray.glover@uk.ibm.com>
Robert Guo <robert.guo@mongodb.com>
Roman Lebedev <lebedev.ri@gmail.com>
Shuo Chen <chenshuo@chenshuo.com>
Tobias Ulvgrd <tobias.ulvgard@dirac.se>
Tom Madams <tom.ej.madams@gmail.com> <tmadams@google.com>
Yixuan Qiu <yixuanq@gmail.com>
Yusuke Suzuki <utatane.tea@gmail.com>
Zbigniew Skowron <zbychs@gmail.com>

# 1.116 libcxxabi 9.0.8svn

## 1.116.1 Available under license :

==========
libc++abi License
==========

The libc++abi library is dual licensed under both the University of Illinois
"BSD-Like" license and the MIT license.  As a user of this code you may choose
to use it under either license.  As a contributor, you agree to allow your code

to be used under both.

Full text of the relevant licenses is included below.

==========

University of Illinois/NCSA
Open Source License

Copyright (c) 2009-2018 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction,
including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimers in the
  documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at
  Urbana-Champaign, nor the names of its contributors may be used to
  endorse or promote products derived from this Software without specific
  prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE

SOFTWARE.

==========

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial
portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

==========

libc++abi License

==========

The libc++abi library is dual licensed under both the University of Illinois
"BSD-Like" license and the MIT license.  As a user of this code you may choose
to use it under either license.  As a contributor, you agree to allow your code
to be used under both.

Full text of the relevant licenses is included below.

==========

University of Illinois/NCSA
Open Source License

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction,
including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==========

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial
portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.117 swagger-compat-spec-parser 1.0.55

## 1.117.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
package io.swagger.models.resourcelisting;

import io.swagger.models.SwaggerBaseModel;

public class ApiInfo extends SwaggerBaseModel {
  private String title = null;
  private String description = null;
  private String termsOfServiceUrl = null;
  private String contact = null;
  private String license = null;
  private String licenseUrl = null;

  public String getTitle() {
    return title;
  }

  public void setTitle(String title) {
    this.title = title;
  }

  public String getDescription() {
    return description;
  }

  public void setDescription(String description) {
    this.description = description;
  }

  public String getTermsOfServiceUrl() {
    return termsOfServiceUrl;
  }

  public void setTermsOfServiceUrl(String termsOfServiceUrl) {
    this.termsOfServiceUrl = termsOfServiceUrl;
```

```
  }

  public String getContact() {
     return contact;
   }

  public void setContact(String contact) {
     this.contact = contact;
  }

  public String getLicense() {
     return license;
  }

  public void setLicense(String license) {
     this.license = license;
  }

  public String getLicenseUrl() {
     return licenseUrl;
  }

  public void setLicenseUrl(String licenseUrl) {
     this.licenseUrl = licenseUrl;
  }

  @Override
  public String toString() {
     StringBuilder sb = new StringBuilder();
     sb.append("class ApiInfo {\n");
     sb.append("  title: ").append(title).append("\n");
     sb.append("  description: ").append(description).append("\n");
     sb.append("  termsOfServiceUrl: ").append(termsOfServiceUrl).append("\n");
     sb.append("  contact: ").append(contact).append("\n");
     sb.append("  license: ").append(license).append("\n");
     sb.append("  licenseUrl: ").append(licenseUrl).append("\n");
     sb.append("  extraFields: ").append(getExtraFields()).append("\n");
      sb.append("}\n");
     return sb.toString();
  }
}
```

Found in path(s):
* /opt/cola/permits/1250771656_1641538233.06/0/swagger-compat-spec-parser-1-0-55-sources-jar/io/swagger/models/resourcelisting/ApiInfo.java

# 1.118 swagger-parser 1.0.55

## 1.118.1 Available under license :

No license file was found, but licenses were detected in source scan.

public License license(ObjectNode node, String location, ParseResult result) {

Found in path(s):
* /opt/cola/permits/1250771649_1641538397.98/0/swagger-parser-1-0-55-sources-jar/io/swagger/parser/util/SwaggerDeserializer.java

# 1.119 swagger-parser-v2-converter 2.0.26

## 1.119.1 Available under license :

Apache-2.0

# 1.120 commons-compress 1.21

## 1.120.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Commons Compress
Copyright 2002-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

---

The files in the package org.apache.commons.compress.archivers.sevenz

were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/),
which has been placed in the public domain:

"LZMA SDK is placed in the public domain." (http://www.7-zip.org/sdk.html)

---

The test file lbzip2_32767.bz2 has been copied from libbzip2's source
repository:

This program, "bzip2", the associated library "libbzip2", and all
documentation, are copyright (C) 1996-2019 Julian R Seward. All
rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented;
 you must
  not claim that you wrote the original software. If you use this
  software in a product, an acknowledgment in the product
  documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must
  not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote
  products derived from this software without specific prior written
  permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER
 IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, jseward@acm.org

# 1.121 checker-qual 3.12.0

## 1.121.1 Available under license :

Checker Framework qualifiers
Copyright 2004-present by the Checker Framework developers

MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

# 1.122 stax-api 4.2.1

## 1.122.1 Available under license :

This copy of Stax2 API is licensed under the
Simplified BSD License (also known as "2-clause BSD", or "FreeBSD License")
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.opensource.org/licenses/bsd-license.php

with details of:

<COPYRIGHT HOLDER> = FasterXML.com
<YEAR> = 2010-

# 1.123 hawtjni-runtime 1.15

## 1.123.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*******************************************************************************
* Copyright (C) 2009-2011 FuseSource Corp.
* Copyright (c) 2004, 2008 IBM Corporation and others.
*
* All rights reserved. This program and the accompanying materials
* are made available under the terms of the Eclipse Public License v1.0
* which accompanies this distribution, and is available at
* http://www.eclipse.org/legal/epl-v10.html
*******************************************************************************/
```

Found in path(s):
* /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/FieldFlag.java
* /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/ClassFlag.java
* /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/ArgFlag.java

No license file was found, but licenses were detected in source scan.

```
/*******************************************************************************
* Copyright (C) 2009-2011 FuseSource Corp.
* Copyright (c) 2000, 2008 IBM Corporation and others.
*
* All rights reserved. This program and the accompanying materials
* are made available under the terms of the Eclipse Public License v1.0
* which accompanies this distribution, and is available at
* http://www.eclipse.org/legal/epl-v10.html
*******************************************************************************/
```

Found in path(s):
* /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/MethodFlag.java

No license file was found, but licenses were detected in source scan.

```
/*******************************************************************************
* Copyright (C) 2009-2011 FuseSource Corp.
* Copyright (c) 2004, 2006 IBM Corporation and others.
*
* All rights reserved. This program and the accompanying materials
* are made available under the terms of the Eclipse Public License v1.0
* which accompanies this distribution, and is available at
* http://www.eclipse.org/legal/epl-v10.html
```

**********/

Found in path(s):
* /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/NativeStats.java
No license file was found, but licenses were detected in source scan.

/**********
* Copyright (c) 2000, 2007 IBM Corporation and others.
* All rights reserved. This program and the accompanying materials
* are made available under the terms of the Eclipse Public License v1.0
* which accompanies this distribution, and is available at
* http://www.eclipse.org/legal/epl-v10.html
*
* Contributors:
*  IBM Corporation - initial API and implementation
**********/

Found in path(s):
* /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/Callback.java
No license file was found, but licenses were detected in source scan.

/**
* Copyright (C) 2010, FuseSource Corp.  All rights reserved.
*/

Found in path(s):
* /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/JNIEnv.java
No license file was found, but licenses were detected in source scan.

/**********
* Copyright (C) 2009-2011 FuseSource Corp.
* Copyright (c) 2000, 2009 IBM Corporation and others.
*
* All rights reserved. This program and the accompanying materials
* are made available under the terms of the Eclipse Public License v1.0
* which accompanies this distribution, and is available at
* http://www.eclipse.org/legal/epl-v10.html
**********/

Found in path(s):
* /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/Library.java
No license file was found, but licenses were detected in source scan.

/**********

* Copyright (C) 2009-2011 FuseSource Corp.
*
* All rights reserved. This program and the accompanying materials
* are made available under the terms of the Eclipse Public License v1.0
* which accompanies this distribution, and is available at
* http://www.eclipse.org/legal/epl-v10.html
**********/

Found in path(s):
* /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/JniField.java
* /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/JniArg.java
* /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/T32.java
* /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/JniClass.java
*
 /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/JniMethod.java
No license file was found, but licenses were detected in source scan.

/**********
* Copyright (C) 2009-2011 FuseSource Corp.
*
* All rights reserved. This program and the accompanying materials
* are made available under the terms of the Eclipse Public License v1.0
* which accompanies this distribution, and is available at
* http://www.eclipse.org/legal/epl-v10.html
**********/

Found in path(s):
* /opt/cola/permits/1257496743_1654022829.8915339/0/hawtjni-runtime-1-15-sources-jar/org/fusesource/hawtjni/runtime/PointerMath.java

# 1.124 jansi-native 1.7

## 1.124.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009-2017 the original author(s).
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*

```
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1257496935_1642830206.82/0/jansi-native-1-7-sources-jar/org/fusesource/jansi/internal/CLibrary.java
* /opt/cola/permits/1257496935_1642830206.82/0/jansi-native-1-7-sources-jar/org/fusesource/jansi/internal/Kernel32.java
* /opt/cola/permits/1257496935_1642830206.82/0/jansi-native-1-7-sources-jar/org/fusesource/jansi/internal/WindowsSupport.java

# 1.125 jansi-osx 1.7

## 1.125.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2009-2017 the original author(s).
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1257496796_1642830208.24/0/jansi-osx-1-7-sources-jar/org/fusesource/jansi/internal/Kernel32.java
* /opt/cola/permits/1257496796_1642830208.24/0/jansi-osx-1-7-sources-jar/org/fusesource/jansi/internal/WindowsSupport.java
* /opt/cola/permits/1257496796_1642830208.24/0/jansi-osx-1-7-sources-jar/org/fusesource/jansi/internal/CLibrary.java

# 1.126 jansi 1.16

## 1.126.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2009-2017 the original author(s).
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1257496944_1642830294.13/0/jansi-1-16-sources-jar/org/fusesource/jansi/Ansi.java
* /opt/cola/permits/1257496944_1642830294.13/0/jansi-1-16-sources-jar/org/fusesource/jansi/AnsiRenderer.java
* /opt/cola/permits/1257496944_1642830294.13/0/jansi-1-16-sources-jar/org/fusesource/jansi/WindowsAnsiOutputStream.java
* /opt/cola/permits/1257496944_1642830294.13/0/jansi-1-16-sources-jar/org/fusesource/jansi/AnsiConsole.java
* /opt/cola/permits/1257496944_1642830294.13/0/jansi-1-16-sources-jar/org/fusesource/jansi/AnsiRenderWriter.java
* /opt/cola/permits/1257496944_1642830294.13/0/jansi-1-16-sources-jar/org/fusesource/jansi/AnsiString.java
* /opt/cola/permits/1257496944_1642830294.13/0/jansi-1-16-sources-jar/org/fusesource/jansi/HtmlAnsiOutputStream.java
* /opt/cola/permits/1257496944_1642830294.13/0/jansi-1-16-sources-jar/org/fusesource/jansi/AnsiOutputStream.java

# 1.127 jansi 1.7

## 1.127.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/**
* Copyright (C) 2009, Progress Software Corporation and/or its
* subsidiaries or affiliates.  All rights reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
```

* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1257496942_1642830289.31/0/jansi-1-7-sources-jar/org/fusesource/jansi/Ansi.java
* /opt/cola/permits/1257496942_1642830289.31/0/jansi-1-7-sources-jar/org/fusesource/jansi/AnsiOutputStream.java
* /opt/cola/permits/1257496942_1642830289.31/0/jansi-1-7-sources-jar/org/fusesource/jansi/AnsiConsole.java
* /opt/cola/permits/1257496942_1642830289.31/0/jansi-1-7-sources-jar/org/fusesource/jansi/WindowsAnsiOutputStream.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 the original author(s).
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1257496942_1642830289.31/0/jansi-1-7-sources-jar/org/fusesource/jansi/AnsiString.java
* /opt/cola/permits/1257496942_1642830289.31/0/jansi-1-7-sources-jar/org/fusesource/jansi/AnsiRenderWriter.java
* /opt/cola/permits/1257496942_1642830289.31/0/jansi-1-7-sources-jar/org/fusesource/jansi/AnsiRenderer.java

# 1.128 gson 2.9.0

## 1.128.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/DateTypeAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/TreeTypeAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/UnsafeAllocator.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/sql/SqlDateTypeAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/ConstructorConstructor.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/LazilyParsedNumber.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/sql/SqlTimeTypeAdapter.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2009 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
```

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonStreamParser.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/FieldAttributes.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonParser.java
*
 /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/LongSerializationPolicy.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Gson authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/JavaVersion.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/PreJava9DateFormatProvider.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/TypeAdapters.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/TypeAdapterRuntimeTypeWrapper.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/JsonTreeReader.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/JsonTreeWriter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/ArrayTypeAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/TypeAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/CollectionTypeAdapterFactory.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/JsonReaderInternalAccess.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/ReflectiveTypeAdapterFactory.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/ObjectTypeAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/TypeAdapterFactory.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/MapTypeAdapterFactory.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Android Open Source Project
* Copyright (C) 2012 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/LinkedTreeMap.java
No license file was found, but licenses were detected in source scan.

/**
* Copyright (C) 2008 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/$Gson$Types.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/stream/MalformedJsonException.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/stream/JsonWriter.java

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/stream/JsonToken.java
*
/opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/stream/JsonReader.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/stream/JsonScope.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Gson authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/GsonBuildConfig.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/ToNumberStrategy.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/ToNumberPolicy.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2010 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/Streams.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonSyntaxException.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2014 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/annotations/JsonAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/JsonAdapterAnnotationTypeAdapterFactory.java
No license file was found, but licenses were detected in source scan.

```
/*
```

* Copyright (C) 2020 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/NumberTypeAdapter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/annotations/Until.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonElement.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/Excluder.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/ExclusionStrategy.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonParseException.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/$Gson$Preconditions.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonIOException.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonObject.java

* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/InstanceCreator.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/Primitives.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonNull.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/annotations/SerializedName.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonDeserializationContext.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/FieldNamingPolicy.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonPrimitive.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/FieldNamingStrategy.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/Gson.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonSerializer.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonSerializationContext.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/annotations/Since.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonDeserializer.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/reflect/TypeToken.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/ObjectConstructor.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/internal/bind/DefaultDateTypeAdapter.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/annotations/Expose.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/GsonBuilder.java
* /opt/cola/permits/1296849109_1648622539.88/0/gson-2-9-0-sources-jar/com/google/gson/JsonArray.java

# 1.129 trove4j 1.0.20200330

## 1.129.1 Available under license :

Found license 'General Public License 2.0' in '// Copyright (c) 2002, Eric D. Friedman All Rights Reserved. // This library is free software; you can redistribute it and/or // modify it under the terms of the GNU Lesser General Public // License as published by the Free Software Foundation; either // version 2.1 of the License, or (at your option) any later version. // This library is distributed in the hope that it will be useful, // but WITHOUT ANY WARRANTY; without even the implied warranty of // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the // GNU General Public License for more details. // You should have received a copy of the GNU Lesser General Public // License along with this program; if not, write to the Free Software'
Found license 'General Public License 2.0' in '// Copyright (c) 2001, Eric D. Friedman All Rights Reserved. // This library is free software; you can redistribute it and/or // modify it under the terms of the GNU Lesser General Public // License as published by the Free Software Foundation; either // version 2.1 of the License, or (at your option) any later version. // This library is distributed in the hope that it will be useful, // but WITHOUT ANY WARRANTY; without even the implied warranty of // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the // GNU General Public License for more details. // You should have received a copy of the GNU Lesser General Public // License along with this program; if not, write to the Free Software'

Found license 'General Public License 2.0' in '// Copyright (c) 2001, Jason Baldridge All Rights Reserved. // This library is free software; you can redistribute it and/or // modify it under the terms of the GNU Lesser General Public // License as published by the Free Software Foundation; either // version 2.1 of the License, or (at your option) any later version. // This library is distributed in the hope that it will be useful, // but WITHOUT ANY WARRANTY; without even the implied warranty of // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the // GNU General Public License for more details. // You should have received a copy of the GNU Lesser General Public // License along with this program; if not, write to the Free Software'

# 1.130 jackson-core 2.12.4

## 1.130.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers.

## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0
To find the details that apply to this artifact see the accompanying LICENSE file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

# 1.131 perfmark-api 0.23.0

## 1.131.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2020 Google LLC
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1302049233_1649313702.49/0/perfmark-api-0-23-0-sources-1-jar/io/perfmark/TaskCloseable.java
* /opt/cola/permits/1302049233_1649313702.49/0/perfmark-api-0-23-0-sources-1-jar/io/perfmark/StringFunction.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2019 Google LLC
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1302049233_1649313702.49/0/perfmark-api-0-23-0-sources-1-jar/io/perfmark/Tag.java

* /opt/cola/permits/1302049233_1649313702.49/0/perfmark-api-0-23-0-sources-1-jar/io/perfmark/package-info.java
* /opt/cola/permits/1302049233_1649313702.49/0/perfmark-api-0-23-0-sources-1-jar/io/perfmark/Link.java
*
 /opt/cola/permits/1302049233_1649313702.49/0/perfmark-api-0-23-0-sources-1-jar/io/perfmark/PerfMark.java
* /opt/cola/permits/1302049233_1649313702.49/0/perfmark-api-0-23-0-sources-1-jar/io/perfmark/Impl.java

# 1.132 grpc-api 1.39.0

## 1.132.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/PartialForwardingServerCall.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/EquivalentAddressGroup.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/CompressorRegistry.java
*
 /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ManagedChannelBuilder.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/Codec.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/Compressor.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ForwardingServerCallListener.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/Contexts.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServerProvider.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ForwardingClientCall.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ForwardingClientCallListener.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServerBuilder.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/Attributes.java
*

/opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ExperimentalApi.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ForwardingServerCall.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/Internal.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/PartialForwardingServerCallListener.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/package-info.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/StatusException.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ManagedChannelProvider.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/NameResolver.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/CallOptions.java
*
/opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/DecompressorRegistry.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ManagedChannel.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/StatusRuntimeException.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/Decompressor.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2020 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ManagedChannelRegistry.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/CompositeChannelCredentials.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ChoiceChannelCredentials.java
*
/opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InsecureServerCredentials.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ChannelCredentials.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServerCredentials.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InsecureChannelCredentials.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalConfigSelector.java

* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/CompositeCallCredentials.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServerRegistry.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ForwardingServerBuilder.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ChoiceServerCredentials.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalServer.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/HasByteBuffer.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2020 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
   * Key managers and trust managers may be specified as {@link KeyManager} and {@link
   * TrustManager} objects. This feature by itself only implies {@link #getKeyManagers()} needs to
   * be observed. But along with {@link #MTLS}, then {@link #getTrustManagers()} needs to be
   * observed as well. When a manager is
 non-{@code null}, then it is wholly responsible for key
   * or trust material and usage; there is no need to check other manager sources like {@link
   * #getCertificateChain()} or {@link #getPrivateKey()} (if {@code KeyManager} is available), or
   * {@link #getRootCertificates()} (if {@code TrustManager} is available).
   *
   * <p>If other manager sources are available (e.g., {@code getPrivateKey() != null}), then they
   * may be alternative representations of the same configuration and the consumer is free to use
   * those alternative representations if it prefers. But before doing so it <em>must</em> first
   * check that it understands that alternative representation by using {@link #incomprehensible}
   * <em>without</em> the {@code CUSTOM_MANAGERS} feature.
   */
```

Found in path(s):
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/TlsServerCredentials.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2020 The gRPC Authors
```

```
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
   * Key managers and trust managers may be specified as {@link KeyManager} and {@link
   * TrustManager} objects. This feature requires observing {@link #getKeyManagers()} and {@link
   * #getTrustManagers()}. Generally {@link #MTLS} should also be supported, as that is the more
   * common method of configuration.
 When a manager is non-{@code null}, then it is wholly
   * responsible for key or trust material and usage; there is no need to check other manager
   * sources like {@link #getCertificateChain()} or {@link #getPrivateKey()} (if {@code
   * KeyManager} is available), or {@link #getRootCertificates()} (if {@code TrustManager} is
   * available).
   *
   * <p>If other manager sources are available (e.g., {@code getPrivateKey() != null}), then they
   * may be alternative representations of the same configuration and the consumer is free to use
   * those alternative representations if it prefers. But before doing so it <em>must</em> first
   * check that it understands that alternative representation by using {@link #incomprehensible}
   * <em>without</em> the {@code CUSTOM_MANAGERS} feature.
   */
```

Found in path(s):
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/TlsChannelCredentials.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2017 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
```

* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ProxyDetector.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalInstrumented.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServiceProviders.java
*
 /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServerStreamTracer.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-
jar/io/grpc/ForwardingChannelBuilder.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalServiceProviders.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalLogId.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/StreamTracer.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalWithLogId.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-
jar/io/grpc/InternalServerInterceptors.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalStatus.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ClientStreamTracer.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2018 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-
jar/io/grpc/PartialForwardingClientCallListener.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/SynchronizationContext.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-
jar/io/grpc/PartialForwardingClientCall.java
*
 /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ChannelLogger.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/LoadBalancerRegistry.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/LoadBalancerProvider.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalClientInterceptors.java

* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalChannelz.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalMetadata.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ConnectivityStateInfo.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalDecompressorRegistry.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServerTransportFilter.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/SecurityLevel.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/BindableService.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServiceDescriptor.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/NameResolverProvider.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/CallCredentials.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/LoadBalancer.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalKnownTransport.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/Grpc.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/CallCredentials2.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ConnectivityState.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalMethodDescriptor.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2018, gRPC Authors All rights reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*

*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/BinaryLog.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/NameResolverRegistry.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ProxiedSocketAddress.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/HttpConnectProxiedSocketAddress.java
*
 /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/InternalCallOptions.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/KnownLength.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServerMethodDefinition.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/Metadata.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServerCallHandler.java
*
 /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ClientInterceptors.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/Server.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServerServiceDefinition.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/Status.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/Drainable.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/Channel.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/HandlerRegistry.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ClientCall.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServerInterceptor.java
*
 /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServerCall.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ClientInterceptor.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/MethodDescriptor.java
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/ServerInterceptors.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2021 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312600_1649732094.9/0/grpc-api-1-39-0-sources-jar/io/grpc/Detachable.java

# 1.133 grpc-core 1.39.0

## 1.133.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ClientStream.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/AbstractServerStream.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/CompositeReadableBuffer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/AbstractStream.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ReadableBuffer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ClientCallImpl.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ServerStream.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/AbstractReadableBuffer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/MessageDeframer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ServerImpl.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ClientTransportFactory.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ClientStreamListener.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/StreamListener.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/TransportFrameUtil.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ServerStreamListener.java

* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ForwardingReadableBuffer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/Http2ClientStreamTransportState.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/AbstractClientStream.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/Stream.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ReadableBuffers.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ServerListener.java
*
 /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/SharedResourceHolder.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/MessageFramer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ServerTransportListener.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/GrpcUtil.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/SerializingExecutor.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/util/MutableHandlerRegistry.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ManagedClientTransport.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/util/RoundRobinLoadBalancer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/AbstractSubchannel.java
*

/opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/SharedResourcePool.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/InternalHandlerRegistry.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/MetadataApplierImpl.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/OobChannel.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ObjectPool.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/KeepAliveManager.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ManagedChannelImpl.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/LogExceptionRunnable.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ClientTransport.java
*
/opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ConnectivityStateManager.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/CallCredentialsApplyingTransportFactory.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/SubchannelChannel.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/InUseStateAggregator.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ForwardingConnectionClientTransport.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ConnectionClientTransport.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/PickSubchannelArgsImpl.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/FailingClientStream.java
*
/opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/StatsTraceContext.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/inprocess/InProcessSocketAddress.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/FailingClientTransport.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2021 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*

```
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/util/CertificateUtils.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2018 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/JsonParser.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/util/SecretRoundRobinLoadBalancerProvider.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/HedgingPolicy.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ForwardingManagedChannel.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/util/ForwardingLoadBalancerHelper.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ChannelTracer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/JndiResourceResolverFactory.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ServiceConfigUtil.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ChannelLoggerImpl.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ServerCallInfoImpl.java

* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ManagedChannelOrphanWrapper.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ForwardingClientStream.java
*
 /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/AutoConfiguredLoadBalancerFactory.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ForwardingClientStreamListener.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/PickFirstLoadBalancerProvider.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/util/ForwardingLoadBalancer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/Rescheduler.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/RetryPolicy.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2019 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/JsonUtil.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ConscryptLoader.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ServiceConfigState.java
*
 /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/util/GracefulSwitchLoadBalancer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/TransportProvider.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/util/ForwardingClientStreamTracer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ManagedChannelServiceConfig.java

* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ThreadOptimizedDeframer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/util/ForwardingSubchannel.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/InsightBuilder.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ApplicationThreadDeframerListener.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ForwardingDeframerListener.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/MigratingThreadDeframer.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/package-info.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/SerializeReentrantCallsDirectExecutor.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ServerTransport.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ServerCallImpl.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/inprocess/InProcessServerBuilder.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/DelayedStream.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/inprocess/InProcessChannelBuilder.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/PickFirstLoadBalancer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/inprocess/InProcessServer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/InternalServer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-

jar/io/grpc/internal/InternalSubchannel.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/Http2Ping.java
*
 /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-
jar/io/grpc/internal/DnsNameResolver.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-
jar/io/grpc/internal/DelayedClientTransport.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/BackoffPolicy.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/WritableBuffer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-
jar/io/grpc/internal/NoopClientStream.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-
jar/io/grpc/internal/ContextRunnable.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-
jar/io/grpc/internal/WritableBufferAllocator.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/inprocess/package-info.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-
jar/io/grpc/internal/ExponentialBackoffPolicy.java
*
 /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-
jar/io/grpc/inprocess/InProcessTransport.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-
jar/io/grpc/internal/DnsNameResolverProvider.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/GrpcAttributes.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/TimeProvider.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/Framer.java
*
 /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/TransportTracer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/RetriableStream.java

* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/CallTracer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/GzipInflatingBuffer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ProxyDetectorImpl.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/LongCounterFactory.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/FixedObjectPool.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/LongCounter.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/AtomicLongCounter.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/util/TransmitStatusRuntimeExceptionInterceptor.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ReflectionLongAdderCounter.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/Deframer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ApplicationThreadDeframer.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ForwardingNameResolver.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/util/package-info.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/AtomicBackoff.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2020 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/DelayedClientCall.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/inprocess/InternalInProcessServerBuilder.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-

jar/io/grpc/internal/SquelchLateMessagesAvailableDeframerListener.java
*
 /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/inprocess/InternalInProcessChannelBuilder.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ServerImplBuilder.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/AbstractServerImplBuilder.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/AbstractManagedChannelImplBuilder.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/internal/ManagedChannelImplBuilder.java
* /opt/cola/permits/1305312581_1649732096.5/0/grpc-core-1-39-0-sources-jar/io/grpc/inprocess/InternalInProcess.java

# 1.134 grpc-stub 1.39.0

## 1.134.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2019 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/AbstractFutureStub.java
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/InternalClientCalls.java
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/AbstractAsyncStub.java
*
 /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/AbstractBlockingStub.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2018 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/annotations/RpcMethod.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2015 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/StreamObservers.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2016 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
```

* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/CallStreamObserver.java
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/ServerCallStreamObserver.java
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/ClientResponseObserver.java
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/ClientCallStreamObserver.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/MetadataUtils.java
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/StreamObserver.java
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/AbstractStub.java
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/ClientCalls.java
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/ServerCalls.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312593_1649732100.9/0/grpc-stub-1-39-0-sources-1-jar/io/grpc/stub/package-info.java

# 1.135 grpc-context 1.39.0

## 1.135.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312579_1649732099.25/0/grpc-context-1-39-0-sources-jar/io/grpc/Deadline.java
* /opt/cola/permits/1305312579_1649732099.25/0/grpc-context-1-39-0-sources-jar/io/grpc/ThreadLocalContextStorage.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright 2017 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312579_1649732099.25/0/grpc-context-1-39-0-sources-jar/io/grpc/PersistentHashArrayMappedTrie.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312579_1649732099.25/0/grpc-context-1-39-0-sources-jar/io/grpc/Context.java

# 1.136 grpc-protobuf-lite 1.39.0

## 1.136.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The gRPC Authors
*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312607_1649732093.16/0/grpc-protobuf-lite-1-39-0-sources-1-jar/io/grpc/protobuf/lite/ProtoInputStream.java
* /opt/cola/permits/1305312607_1649732093.16/0/grpc-protobuf-lite-1-39-0-sources-1-jar/io/grpc/protobuf/lite/ProtoLiteUtils.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312607_1649732093.16/0/grpc-protobuf-lite-1-39-0-sources-1-jar/io/grpc/protobuf/lite/package-info.java

# 1.137 grpc-protobuf 1.39.0

## 1.137.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The gRPC Authors

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312572_1649732103.09/0/grpc-protobuf-1-39-0-sources-1-jar/io/grpc/protobuf/ProtoUtils.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312572_1649732103.09/0/grpc-protobuf-1-39-0-sources-1-jar/io/grpc/protobuf/ProtoServiceDescriptorSupplier.java
* /opt/cola/permits/1305312572_1649732103.09/0/grpc-protobuf-1-39-0-sources-1-jar/io/grpc/protobuf/ProtoMethodDescriptorSupplier.java
* /opt/cola/permits/1305312572_1649732103.09/0/grpc-protobuf-1-39-0-sources-1-jar/io/grpc/protobuf/package-info.java
*
 /opt/cola/permits/1305312572_1649732103.09/0/grpc-protobuf-1-39-0-sources-1-jar/io/grpc/protobuf/StatusProto.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright 2016 The gRPC Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1305312572_1649732103.09/0/grpc-protobuf-1-39-0-sources-1-jar/io/grpc/protobuf/ProtoFileDescriptorSupplier.java

# 1.138 kotlinx-coroutines-core-jvm 1.5.0

## 1.138.1 Available under license :

Apache-2.0

# 1.139 failureaccess 1.0.1

## 1.139.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1349013147_1655879756.6643403/0/failureaccess-1-0-1-sources-4-

jar/com/google/common/util/concurrent/internal/InternalFutures.java
* /opt/cola/permits/1349013147_1655879756.6643403/0/failureaccess-1-0-1-sources-4-jar/com/google/common/util/concurrent/internal/InternalFutureFailureAccess.java

# 1.140 error_prone_annotations 2.3.4

## 1.140.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2016 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/RestrictedApi.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/MustBeClosed.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/CompatibleWith.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/DoNotMock.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/FormatMethod.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/FormatString.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2017 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
```

*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/concurrent/GuardedBy.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/CheckReturnValue.java
*
 /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/DoNotCall.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/OverridingMethodsMustInvokeSuper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/concurrent/LazyInit.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/Immutable.java
*
 /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/ForOverride.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/IncompatibleModifiers.java

* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/CanIgnoreReturnValue.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/CompileTimeConstant.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/RequiredModifiers.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/Var.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/SuppressPackageLocation.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/concurrent/UnlockMethod.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/concurrent/LockMethod.java
* /opt/cola/permits/1349040975_1655882556.5391665/0/error-prone-annotations-2-3-4-sources-11-jar/com/google/errorprone/annotations/NoAllocation.java

# 1.141 listenablefuture 9999.0-empty-to-avoid-conflict-with-guava

## 1.141.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.142 adapter-rxjava3 2.9.0

## 1.142.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020  Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1367722265_1658240257.7390077/0/adapter-rxjava3-2-9-0-sources-jar/retrofit2/adapter/rxjava3/RxJava3CallAdapter.java
* /opt/cola/permits/1367722265_1658240257.7390077/0/adapter-rxjava3-2-9-0-sources-jar/retrofit2/adapter/rxjava3/ResultObservable.java
* /opt/cola/permits/1367722265_1658240257.7390077/0/adapter-rxjava3-2-9-0-sources-jar/retrofit2/adapter/rxjava3/CallEnqueueObservable.java
*
 /opt/cola/permits/1367722265_1658240257.7390077/0/adapter-rxjava3-2-9-0-sources-jar/retrofit2/adapter/rxjava3/BodyObservable.java
* /opt/cola/permits/1367722265_1658240257.7390077/0/adapter-rxjava3-2-9-0-sources-

jar/retrofit2/adapter/rxjava3/CallExecuteObservable.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1367722265_1658240257.7390077/0/adapter-rxjava3-2-9-0-sources-jar/retrofit2/adapter/rxjava3/Result.java
* /opt/cola/permits/1367722265_1658240257.7390077/0/adapter-rxjava3-2-9-0-sources-jar/retrofit2/adapter/rxjava3/HttpException.java
* /opt/cola/permits/1367722265_1658240257.7390077/0/adapter-rxjava3-2-9-0-sources-jar/retrofit2/adapter/rxjava3/RxJava3CallAdapterFactory.java

# 1.143 logging-interceptor 4.10.0

## 1.143.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2015 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1392932605_1660763047.3029058/0/logging-interceptor-4-10-0-sources-jar/okhttp3/logging/HttpLoggingInterceptor.kt
* /opt/cola/permits/1392932605_1660763047.3029058/0/logging-interceptor-4-10-0-sources-jar/okhttp3/logging/utf8.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2018 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1392932605_1660763047.3029058/0/logging-interceptor-4-10-0-sources-jar/okhttp3/logging/LoggingEventListener.kt

# 1.144 flatbuffers-java 1.12.0

## 1.144.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
// Copyright 2008 Google Inc.  All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
//     * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//     * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
//     * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.
```

Found in path(s):
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/Utf8Safe.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2019 Google Inc. All rights reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/UnionVector.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/LongVector.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/ByteVector.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/IntVector.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/BooleanVector.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/DoubleVector.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/FloatVector.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/ShortVector.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/BaseVector.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/StringVector.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 Google Inc. All rights reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
```

*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/FlexBuffersBuilder.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/Constants.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/Struct.java
*
 /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/Table.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/FlatBufferBuilder.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/FlexBuffers.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/Utf8.java
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-jar/com/google/flatbuffers/Utf8Old.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 Google Inc. All rights reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1421758158_1663678673.102127/0/flatbuffers-java-1-12-0-sources-1-

jar/com/google/flatbuffers/ByteBufferUtil.java

# 1.145 gson 2.8.9

## 1.145.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonPrimitive.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonDeserializationContext.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/Primitives.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/annotations/Expose.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/GsonBuilder.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonNull.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonSerializationContext.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/InstanceCreator.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonParseException.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonObject.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/FieldNamingStrategy.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonArray.java

*
 /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/$Gson$Preconditions.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/annotations/Since.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/ObjectConstructor.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonElement.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/Excluder.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/ExclusionStrategy.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonSerializer.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/annotations/Until.java
*
 /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/FieldNamingPolicy.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/reflect/TypeToken.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/Gson.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/annotations/SerializedName.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/bind/DefaultDateTypeAdapter.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonIOException.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonDeserializer.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/ToNumberPolicy.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/ToNumberStrategy.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/bind/NumberTypeAdapter.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2010 The Android Open Source Project
* Copyright (C) 2012 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-

jar/com/google/gson/internal/LinkedTreeMap.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-
jar/com/google/gson/internal/LinkedHashTreeMap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-
jar/com/google/gson/internal/sql/SqlTimeTypeAdapter.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-
jar/com/google/gson/internal/UnsafeAllocator.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-
jar/com/google/gson/internal/bind/DateTypeAdapter.java
*
 /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-
jar/com/google/gson/internal/bind/TreeTypeAdapter.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-
jar/com/google/gson/internal/ConstructorConstructor.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-
jar/com/google/gson/internal/sql/SqlDateTypeAdapter.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-
jar/com/google/gson/internal/LazilyParsedNumber.java
No license file was found, but licenses were detected in source scan.

/**
* Copyright (C) 2008 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/$Gson$Types.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/LongSerializationPolicy.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonParser.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/FieldAttributes.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonStreamParser.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/JsonReaderInternalAccess.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/bind/JsonTreeReader.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/bind/CollectionTypeAdapterFactory.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/TypeAdapter.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/bind/ReflectiveTypeAdapterFactory.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/bind/TypeAdapters.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/bind/ObjectTypeAdapter.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/bind/ArrayTypeAdapter.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/bind/MapTypeAdapterFactory.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/bind/JsonTreeWriter.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/TypeAdapterFactory.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/bind/TypeAdapterRuntimeTypeWrapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/JsonSyntaxException.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/Streams.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/stream/JsonToken.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/stream/JsonScope.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/stream/JsonWriter.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/stream/JsonReader.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/stream/MalformedJsonException.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*

*	http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/annotations/JsonAdapter.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/bind/JsonAdapterAnnotationTypeAdapterFactory.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Gson authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/JavaVersion.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/reflect/UnsafeReflectionAccessor.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/reflect/ReflectionAccessor.java
*
 /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/reflect/PreJava9ReflectionAccessor.java
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/PreJava9DateFormatProvider.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Gson authors

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1550776944_1675301335.8363905/0/gson-2-8-9-sources-4-jar/com/google/gson/internal/GsonBuildConfig.java

# 1.146 jimfs 1.1

## 1.146.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.147 okhttp 4.10.0

## 1.147.1 Available under license :

Note that publicsuffixes.gz is compiled from The Public Suffix List:
https://publicsuffix.org/list/public_suffix_list.dat

It is subject to the terms of the Mozilla Public License, v. 2.0:
https://mozilla.org/MPL/2.0/

# 1.148 jose4j 0.7.0

## 1.148.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements.  See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License.  You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/base64url/internal/apache/commons/codec/binary/Base64.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/base64url/internal/apache/commons/codec/binary/BaseNCodec.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2012-2018 Brian Campbell
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*    http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
```

*/

Found in path(s):
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/HttpsJwks.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012-2017 Brian Campbell
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/RsaJsonWebKey.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/ReservedClaimNames.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/keys/HmacKey.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/KeyManagementAlgorithmIdentifiers.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwx/HeaderParameterNames.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/EllipticCurveJsonWebKey.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/EcdhKeyAgreementWithAesKeyWrapAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwa/AlgorithmFactory.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwx/JsonWebStructure.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/mac/MacUtil.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/IntegrityException.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/BouncyCastleProviderHelp.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/kdf/KdfUtil.java

* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/base64url/SimplePEMEncoder.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/UnresolvableKeyException.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/consumer/IssValidator.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/keys/X509Util.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/consumer/ErrorCodeValidatorAdapter.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/EcJwkGenerator.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwx/CompactSerializer.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/CipherUtil.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jws/PlaintextNoneAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwx/KeyValidationSupport.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/base64url/Base64Url.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jws/JsonWebSignature.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/consumer/AudValidator.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/zip/CompressionAlgorithmIdentifiers.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/consumer/SubValidator.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/JsonWebKeySet.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/keys/PbkdfKey.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/AesKeyWrapManagementAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwa/Algorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/ContentEncryptionHelp.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/JsonHelp.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jws/EcdsaUsingShaAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/ContentEncryptionParts.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/keys/resolvers/VerificationKeyResolver.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/JsonWebKey.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-

jar/org/jose4j/jwt/consumer/JwtContext.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/IntDate.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwe/Pbes2HmacShaWithAesKeyWrapAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwe/ContentEncryptionAlgorithmIdentifiers.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwt/consumer/InvalidJwtException.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/keys/resolvers/JwksVerificationKeyResolver.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwa/AlgorithmConstraints.java
*
/opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/zip/CompressionAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwe/RsaKeyManagementAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/keys/resolvers/HttpsJwksVerificationKeyResolver.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwk/VerificationJwkSelector.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwe/KeyManagementAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwe/ContentEncryptionKeys.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwa/AlgorithmInfo.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwe/EcdhKeyAgreementAlgorithm.java
*
/opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwe/AesCbcHmacSha2ContentEncryptionAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/ByteUtil.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwt/consumer/JwsCustomizer.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwe/WrappingKeyManagementAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/zip/DeflateRFC1951CompressionAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwe/CipherStrengthSupport.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/keys/RsaKeyUtil.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-
jar/org/jose4j/jwt/consumer/JwtConsumerBuilder.java
*
/opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/json/JsonUtil.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-

jar/org/jose4j/jwt/consumer/Validator.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jca/ProviderContext.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/keys/KeyPairUtil.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/consumer/SimpleKeyResolver.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/keys/KeyPersuasion.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/StringUtil.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/AesGcmKeyEncryptionAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/keys/EcKeyUtil.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/ExceptionHelp.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/ContentEncryptionKeyDescriptor.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/consumer/JwtConsumer.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/keys/resolvers/X509VerificationKeyResolver.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/OctJwkGenerator.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/http/Get.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/consumer/ErrorCodeValidator.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/ByteGenerator.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/AesGcmContentEncryptionAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jws/JsonWebSignatureAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/DefaultByteGenerator.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/consumer/JtiValidator.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/keys/AesKey.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/NumericDate.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/JoseException.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/GeneralJwtException.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jws/RsaUsingShaAlgorithm.java

* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/keys/BigEndianBigInteger.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/ContentEncryptionAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/keys/resolvers/DecryptionKeyResolver.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwx/Headers.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jws/HmacUsingShaAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jws/BaseSignatureAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/PublicJsonWebKey.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/OctetSequenceJsonWebKey.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/consumer/JweCustomizer.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/UncheckedJoseException.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/keys/EllipticCurves.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/MalformedClaimException.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/JwtClaims.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/JsonWebEncryption.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/kdf/PasswordBasedKeyDerivationFunction2.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/http/SimpleGet.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/consumer/ErrorCodes.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/InitializationVectorHelp.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwt/consumer/NumericDateValidator.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/DirectKeyManagementAlgorithm.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwa/AlgorithmFactoryFactory.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/InvalidAlgorithmException.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwa/AlgorithmAvailability.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwx/CompactSerialization.java
*

/opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/Use.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/SimpleAeadCipher.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jws/AlgorithmIdentifiers.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/RsaJwkGenerator.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/HashUtil.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/KeyOperations.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/json/JsonHeaderUtil.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/keys/resolvers/JwksDecryptionKeyResolver.java
*
/opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwe/kdf/ConcatKeyDerivationFunction.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012-2017 Brian Campbell
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/SimpleJwkFilter.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/DecryptionJwkSelector.java
* /opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/jwk/SelectorSupport.java
*
/opt/cola/permits/1660960880_1683014432.3431854/0/jose4j-0-7-0-sources-2-jar/org/jose4j/lang/InvalidKeyException.java

# 1.149 okio-jvm 3.0.0

## 1.149.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/HashingSink.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/Options.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/HashingSource.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/Pipe.kt

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2019 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/internal/-Buffer.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/BufferedSource.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/Okio.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/Source.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/RealBufferedSource.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/RealBufferedSink.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/Timeout.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/BufferedSink.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/Buffer.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/Sink.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements.  See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License.  You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/internal/ZipEntry.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/ZipFileSystem.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/internal/zip.kt
No license file was found, but licenses were detected in source scan.

```
/*
```

* Copyright (C) 2017 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/Utf8.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/HashingSink.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/JvmSystemFileSystem.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/FileMetadata.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/Path.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/HashingSource.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/internal/-Path.kt

* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/NioSystemFileSystem.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/FileSystem.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/ForwardingFileSystem.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/ExperimentalFileSystem.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/Path.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2018 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/Throttler.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/-DeprecatedUpgrade.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/-DeprecatedUtf8.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/-DeprecatedOkio.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/PeekSource.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 Square, Inc. and others.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
```

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/CipherSink.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/CipherSource.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 Square Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/ByteString.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/SegmentedByteString.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/SegmentedByteString.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/ForwardingTimeout.kt
No license file was found, but licenses were detected in source scan.

/*
*  Licensed to the Apache Software Foundation (ASF) under one or more
*  contributor license agreements.  See the NOTICE file distributed with
*  this work for additional information regarding copyright ownership.
*  The ASF licenses this file to You under the Apache License, Version 2.0
*  (the "License"); you may not use this file except in compliance with
*  the License.  You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
*  Unless required by applicable law or agreed to in writing, software
*  distributed under the License is distributed on an "AS IS" BASIS,
*  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
*  See the License for the specific language governing permissions and
*  limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/-Base64.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/internal/FixedLengthSource.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/FileHandle.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/JvmFileHandle.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/FileSystem.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/internal/ResourceFileSystem.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/internal/-FileSystem.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2014 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/Timeout.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/Segment.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/SegmentPool.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/ForwardingSink.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/Buffer.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/GzipSource.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/RealBufferedSink.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/AsyncTimeout.kt

* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/InflaterSource.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/DeflaterSink.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/RealBufferedSource.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/SegmentPool.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/BufferedSource.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/JvmOkio.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/BufferedSink.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/ForwardingSource.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/GzipSink.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/Sink.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2019 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/internal/-RealBufferedSource.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/internal/-SegmentedByteString.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/internal/-RealBufferedSink.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2018 Square, Inc.
*
```

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/internal/-ByteString.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/-CommonPlatform.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/jvmMain/okio/-JvmPlatform.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/-Util.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/ByteString.kt
* /opt/cola/permits/1661535429_1683049853.7594967/0/okio-jvm-3-0-0-sources-1-jar/commonMain/okio/internal/-Utf8.kt

# 1.150 xz 1.9

## 1.150.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* CloseIgnoringInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/CloseIgnoringInputStream.java
No license file was found, but licenses were detected in source scan.

/*

* DeltaEncoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/DeltaEncoder.java
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/delta/DeltaEncoder.java
No license file was found, but licenses were detected in source scan.

/*
* RangeCoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*  Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/rangecoder/RangeCoder.java
No license file was found, but licenses were detected in source scan.

/*
* State
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*  Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lzma/State.java
No license file was found, but licenses were detected in source scan.

/*
* LZMACoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*  Igor Pavlov <http://7-zip.org/>
*

* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lzma/LZMACoder.java
No license file was found, but licenses were detected in source scan.

/*
* FilterOptions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*  Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/FilterOptions.java
No license file was found, but licenses were detected in source scan.

/*
* BCJDecoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/BCJDecoder.java
No license file was found, but licenses were detected in source scan.

/*
* Hash Chain match finder with 2-, 3-, and 4-byte hashing
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*  Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lz/HC4.java

No license file was found, but licenses were detected in source scan.

```
/*
* ARMThumbOptions
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/ARMThumbOptions.java
No license file was found, but licenses were detected in source scan.

```
/*
* LZMAEncoderFast
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lzma/LZMAEncoderFast.java
No license file was found, but licenses were detected in source scan.

```
/*
* DeltaOutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/DeltaOutputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* BCJEncoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
```

*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/BCJEncoder.java
No license file was found, but licenses were detected in source scan.

/*
* LZMA2Coder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/LZMA2Coder.java
No license file was found, but licenses were detected in source scan.

/*
* SingleXZInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/SingleXZInputStream.java
No license file was found, but licenses were detected in source scan.

/*
* Optimum
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*  Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lzma/Optimum.java

No license file was found, but licenses were detected in source scan.

```
/*
* LZMA2OutputStream
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/LZMA2OutputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* BCJOptions
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/BCJOptions.java
No license file was found, but licenses were detected in source scan.

```
/*
* DecoderUtil
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/common/DecoderUtil.java
No license file was found, but licenses were detected in source scan.

```
/*
* LZMADecoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
```

```
*         Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lzma/LZMADecoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* 2-, 3-, and 4-byte hashing
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*         Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lz/Hash234.java
No license file was found, but licenses were detected in source scan.

```
/*
* CountingOutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/CountingOutputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* MemoryLimitException
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/MemoryLimitException.java
No license file was found, but licenses were detected in source scan.

```
/*
* LZMA2Encoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/LZMA2Encoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* FinishableOutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/FinishableOutputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* BCJ filter for Itanium (IA-64) instructions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/simple/IA64.java
No license file was found, but licenses were detected in source scan.

```
/*
* LZMAInputStream
```

```
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/LZMAInputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* LZEncoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lz/LZEncoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* XZFormatException
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/XZFormatException.java
No license file was found, but licenses were detected in source scan.

```
/*
* XZOutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/XZOutputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* BlockInfo
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/index/BlockInfo.java
No license file was found, but licenses were detected in source scan.

```
/*
* LZMAEncoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lzma/LZMAEncoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* DeltaCoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/DeltaCoder.java
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/delta/DeltaCoder.java
No license file was found, but licenses were detected in source scan.

```
/*
```

```
* X86Options
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/X86Options.java
No license file was found, but licenses were detected in source scan.

```
/*
* package-info
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
/**
* XZ data compression support.
*
* <h2>Introduction</h2>
* <p>
* This aims to be a complete implementation of XZ data compression
* in pure Java. Features:
* <ul>
* <li>Full support for the .xz file format specification version 1.0.4</li>
* <li>Single-threaded streamed compression and decompression</li>
* <li>Single-threaded decompression with limited random access support</li>
* <li>Raw streams (no .xz headers) for advanced users, including LZMA2
*     with preset dictionary</li>
* </ul>
* <p>
* Threading is planned but it is unknown when it will be implemented.
* <p>
* For the latest source code, see the
* <a href="https://tukaani.org/xz/java.html">home page of XZ for Java</a>.
*
* <h2>Getting started</h2>
* <p>
*
 Start by reading the documentation of {@link org.tukaani.xz.XZOutputStream}
* and {@link org.tukaani.xz.XZInputStream}.
* If you use XZ inside another file format or protocol,
* see also {@link org.tukaani.xz.SingleXZInputStream}.
*
```

```
* <h2>Licensing</h2>
* <p>
* XZ for Java has been put into the public domain, thus you can do
* whatever you want with it. All the files in the package have been
* written by Lasse Collin, Igor Pavlov, and/or Brett Okken.
* <p>
* This software is provided "as is", without any warranty.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/package-info.java
No license file was found, but licenses were detected in source scan.

```
/*
* PowerPCOptions
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/PowerPCOptions.java
No license file was found, but licenses were detected in source scan.

```
/*
* UnsupportedOptionsException
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/UnsupportedOptionsException.java
No license file was found, but licenses were detected in source scan.

```
/*
* LZMAOutputStream
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
```

*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/LZMAOutputStream.java
No license file was found, but licenses were detected in source scan.

/*
* DeltaInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/DeltaInputStream.java
No license file was found, but licenses were detected in source scan.

/*
* LZMA2Options
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/LZMA2Options.java
No license file was found, but licenses were detected in source scan.

/*
* IndexDecoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/index/IndexDecoder.java
No license file was found, but licenses were detected in source scan.

/*

```
* RangeEncoderToBuffer
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/rangecoder/RangeEncoderToBuffer.java
No license file was found, but licenses were detected in source scan.

```
/*
* XZIOException
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/XZIOException.java
No license file was found, but licenses were detected in source scan.

```
/*
* Binary Tree match finder with 2-, 3-, and 4-byte hashing
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lz/BT4.java
No license file was found, but licenses were detected in source scan.

```
/*
* FilterDecoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
```

```
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/FilterDecoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* XZ
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/XZ.java
No license file was found, but licenses were detected in source scan.

```
/*
* CRC32Hash
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lz/CRC32Hash.java
No license file was found, but licenses were detected in source scan.

```
/*
* LZMA2Decoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/LZMA2Decoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* BCJ filter for little endian ARM instructions
```

*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/simple/SimpleFilter.java
No license file was found, but licenses were detected in source scan.

/*
* RangeDecoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*  Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/rangecoder/RangeDecoder.java
No license file was found, but licenses were detected in source scan.

/*
* StreamFlags
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/common/StreamFlags.java
No license file was found, but licenses were detected in source scan.

/*
* LZMA2InputStream
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*  Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.

```
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/LZMA2InputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* BCJ filter for little endian ARM-Thumb instructions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/simple/ARMThumb.java
No license file was found, but licenses were detected in source scan.

```
/*
* DeltaDecoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/delta/DeltaDecoder.java
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/DeltaDecoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* BCJCoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/BCJCoder.java

No license file was found, but licenses were detected in source scan.

```
/*
* LZDecoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lz/LZDecoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* RangeEncoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/rangecoder/RangeEncoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* LZMAEncoderNormal
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lzma/LZMAEncoderNormal.java
No license file was found, but licenses were detected in source scan.

```
/*
* BlockInputStream
```

```
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/BlockInputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* CountingInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/CountingInputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* UncompressedLZMA2OutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/UncompressedLZMA2OutputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* CorruptedInputException
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/CorruptedInputException.java
No license file was found, but licenses were detected in source scan.

```
/*
* SPARCOptions
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/SPARCOptions.java
No license file was found, but licenses were detected in source scan.

```
/*
* SeekableInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/SeekableInputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* Matches
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/lz/Matches.java
No license file was found, but licenses were detected in source scan.

```
/*
* XZInputStream
```

```
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/XZInputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* RawCoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/RawCoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* IndexEncoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/index/IndexEncoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* BCJ filter for x86 instructions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/simple/X86.java
No license file was found, but licenses were detected in source scan.

```
/*
* BCJ filter for big endian PowerPC instructions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/simple/PowerPC.java
No license file was found, but licenses were detected in source scan.

```
/*
* BasicArrayCache
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/BasicArrayCache.java
No license file was found, but licenses were detected in source scan.

```
/*
* ArrayCache
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/ArrayCache.java
No license file was found, but licenses were detected in source scan.

```
/*
* BCJ filter for SPARC instructions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
```

*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/simple/SPARC.java
No license file was found, but licenses were detected in source scan.

/*
* IndexHash
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/index/IndexHash.java
No license file was found, but licenses were detected in source scan.

/*
* None
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/check/None.java
No license file was found, but licenses were detected in source scan.

/*
* DeltaOptions
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/DeltaOptions.java
No license file was found, but licenses were detected in source scan.

```
/*
* Util
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/common/Util.java
No license file was found, but licenses were detected in source scan.

```
/*
* EncoderUtil
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/common/EncoderUtil.java
No license file was found, but licenses were detected in source scan.

```
/*
* ResettableArrayCache
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/ResettableArrayCache.java
No license file was found, but licenses were detected in source scan.

```
/*
* BCJ filter for little endian ARM instructions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
```

* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/simple/ARM.java
No license file was found, but licenses were detected in source scan.

/*
* FilterEncoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/FilterEncoder.java
No license file was found, but licenses were detected in source scan.

/*
* CRC64
*
* Authors: Brett Okken <brett.okken.os@gmail.com>
*  Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/check/CRC64.java
No license file was found, but licenses were detected in source scan.

/*
* IndexIndicatorException
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/IndexIndicatorException.java
No license file was found, but licenses were detected in source scan.

/*
* SHA256
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/check/SHA256.java
No license file was found, but licenses were detected in source scan.

/*
* RangeDecoderFromStream
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*  Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/rangecoder/RangeDecoderFromStream.java
No license file was found, but licenses were detected in source scan.

/*
* IA64Options
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/IA64Options.java
No license file was found, but licenses were detected in source scan.

/*
* ARMOptions
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.

```
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/ARMOptions.java
No license file was found, but licenses were detected in source scan.

```
/*
* IndexBase
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/index/IndexBase.java
No license file was found, but licenses were detected in source scan.

```
/*
* BlockOutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/BlockOutputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* RangeDecoderFromBuffer
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/rangecoder/RangeDecoderFromBuffer.java
No license file was found, but licenses were detected in source scan.

```
/*
```

* SimpleOutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/SimpleOutputStream.java
No license file was found, but licenses were detected in source scan.

/*
* SeekableXZInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/SeekableXZInputStream.java
No license file was found, but licenses were detected in source scan.

/*
* SeekableFileInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/SeekableFileInputStream.java
No license file was found, but licenses were detected in source scan.

/*
* module-info
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.

```
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/META-INF/versions/9/module-info.java
No license file was found, but licenses were detected in source scan.

```
/*
* SimpleInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/SimpleInputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* RangeEncoderToStream
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/rangecoder/RangeEncoderToStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* CRC32
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/check/CRC32.java
No license file was found, but licenses were detected in source scan.

```
/*
* Check
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/check/Check.java
No license file was found, but licenses were detected in source scan.

```
/*
* IndexRecord
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/index/IndexRecord.java
No license file was found, but licenses were detected in source scan.

```
/*
* FilterCoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/FilterCoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* FinishableWrapperOutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/1743261437_1689175256.7452936/0/xz-1-9-sources-jar/org/tukaani/xz/FinishableWrapperOutputStream.java

# 1.151 kotlin-stdlib-jdk8 1.5.31

## 1.151.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1778499234_1692953459.883195/0/kotlin-stdlib-jdk8-1-5-31-sources-2-jar/kotlin/text/RegexExtensions.kt
* /opt/cola/permits/1778499234_1692953459.883195/0/kotlin-stdlib-jdk8-1-5-31-sources-2-jar/kotlin/streams/Streams.kt
* /opt/cola/permits/1778499234_1692953459.883195/0/kotlin-stdlib-jdk8-1-5-31-sources-2-jar/kotlin/collections/Collections.kt
* /opt/cola/permits/1778499234_1692953459.883195/0/kotlin-stdlib-jdk8-1-5-31-sources-2-jar/kotlin/internal/jdk8/JDK8PlatformImplementations.kt

# 1.152 kotlin-stdlib-jdk7 1.5.31

## 1.152.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
```

* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1778499235_1692953555.9059699/0/kotlin-stdlib-jdk7-1-5-31-sources-2-jar/kotlin/AutoCloseable.kt

# 1.153 jsonpatch 1.13

## 1.153.1 Available under license :

Copyright (c) 2011 Stefan Kgl <stefan@skoegl.net>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
  derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL,
 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1.154 jackson-coreutils 2.0

## 1.154.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by
the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters,
data structure
layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.
This software is dual-licensed under:

- the Lesser General Public License (LGPL) version 3.0 or, at your option, any later version;
- the Apache Software License (ASL) version 2.0.

The text of both licenses is included (under the names LGPL-3.0.txt and ASL-2.0.txt respectively).

Direct link to the sources:

- LGPL 3.0: https://www.gnu.org/licenses/lgpl-3.0.txt
- ASL 2.0: http://www.apache.org/licenses/LICENSE-2.0.txt

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.155 swagger-models 1.6.2

## 1.155.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
package io.swagger.models;

import com.fasterxml.jackson.annotation.JsonAnyGetter;
import com.fasterxml.jackson.annotation.JsonAnySetter;

import java.util.LinkedHashMap;
import java.util.Map;

public class License {
  private Map<String, Object> vendorExtensions = new LinkedHashMap<String, Object>();
  private String name;
  private String url;

  public License name(String name) {
      setName(name);
      return this;
  }

  public License url(String url) {
      setUrl(url);
      return this;
  }

  public String getName() {
      return name;
  }
```

```
  public void setName(String name) {
      this.name = name;
  }

  public String getUrl() {
      return url;
  }

  public void setUrl(String url) {
      this.url = url;
  }

  @JsonAnyGetter
  public Map<String, Object> getVendorExtensions() {
      return vendorExtensions;
  }

  @JsonAnySetter
  public void
setVendorExtension(String name, Object value) {
      if (name.startsWith("x-")) {
          vendorExtensions.put(name, value);
      }
  }

  public void setVendorExtensions(Map<String, Object> vendorExtensions) {
      this.vendorExtensions = vendorExtensions;
  }

  @Override
  public int hashCode() {
      final int prime = 31;
      int result = 1;
      result = prime * result + ((name == null) ? 0 : name.hashCode());
      result = prime * result + ((url == null) ? 0 : url.hashCode());
      result = prime * result + ((vendorExtensions == null) ? 0 : vendorExtensions.hashCode());
      return result;
  }

  @Override
  public boolean equals(Object obj) {
      if (this == obj) {
          return true;
      }
      if (obj == null) {
          return false;
      }
      if (getClass() != obj.getClass()) {
```

```
            return false;
        }
        License other = (License) obj;
        if (name == null) {
            if (other.name
 != null) {
                return false;
            }
        } else if (!name.equals(other.name)) {
            return false;
        }
        if (url == null) {
            if (other.url != null) {
                return false;
            }
        } else if (!url.equals(other.url)) {
            return false;
        }
        if (vendorExtensions == null) {
            if (other.vendorExtensions != null) {
                return false;
            }
        } else if (!vendorExtensions.equals(other.vendorExtensions)) {
            return false;
        }
        return true;
    }
}
```

Found in path(s):
* /opt/cola/permits/1861058280_1701797798.5796204/0/swagger-models-1-6-2-sources-3-jar/io/swagger/models/License.java
No license file was found, but licenses were detected in source scan.

```
package io.swagger.models;

import com.fasterxml.jackson.annotation.JsonAnyGetter;
import com.fasterxml.jackson.annotation.JsonAnySetter;

import java.util.LinkedHashMap;
import java.util.Map;

public class Info {
    private String description;
    private String version;
    private String title;
    private String termsOfService;
    private Contact contact;
```

```
private License license;
private Map<String, Object> vendorExtensions = new LinkedHashMap<String, Object>();

public Info version(String version) {
    this.setVersion(version);
    return this;
}

public Info title(String title) {
    this.setTitle(title);
    return this;
}

public Info description(String description) {
    this.setDescription(description);
    return this;
}

public Info termsOfService(String termsOfService) {
    this.setTermsOfService(termsOfService);
    return this;

}

public Info contact(Contact contact) {
    this.setContact(contact);
    return this;
}

public Info license(License license) {
    this.setLicense(license);
    return this;
}

public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

public String getVersion() {
    return version;
}

public void setVersion(String version) {
    this.version = version;
```

```
}

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}

public String getTermsOfService() {
    return termsOfService;
}

public void setTermsOfService(String termsOfService) {
    this.termsOfService = termsOfService;
}

public Contact getContact() {
    return contact;
}

public void setContact(Contact contact) {
    this.contact = contact;
 }

public License getLicense() {
    return license;
}

public void setLicense(License license) {
    this.license = license;
}

public Info mergeWith(Info info) {
    if (info != null) {
        if (this.description == null) {
            this.description = info.description;
        }
        if (this.version == null) {
            this.version = info.version;
        }
        if (this.title == null) {
            this.title = info.title;
        }
        if (this.termsOfService == null) {
            this.termsOfService = info.termsOfService;
        }
```

```
        if (this.contact == null) {
            this.contact = info.contact;
        }
        if (this.license == null) {
            this.license = info.license;
        }
        if (this.vendorExtensions == null) {
            this.vendorExtensions = info.vendorExtensions;
        }
    }
    return this;
 }

 @JsonAnyGetter
 public Map<String, Object> getVendorExtensions() {
    return vendorExtensions;
 }

 @JsonAnySetter
 public void setVendorExtension(String name, Object value) {
    if (name.startsWith("x-")) {
        vendorExtensions.put(name, value);
    }
 }

public void setVendorExtensions(Map<String, Object> vendorExtensions) {
    this.vendorExtensions = vendorExtensions;
 }

 @Override
 public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((contact == null) ? 0 : contact.hashCode());
    result = prime * result
            + ((description == null) ? 0 : description.hashCode());
    result = prime * result + ((license == null) ? 0 : license.hashCode());
    result = prime * result
            + ((termsOfService == null) ? 0 : termsOfService.hashCode());
    result = prime * result + ((title == null) ? 0 : title.hashCode());
    result
= prime * result
            + ((vendorExtensions == null) ? 0 : vendorExtensions.hashCode());
    result = prime * result + ((version == null) ? 0 : version.hashCode());
    return result;
 }

 @Override
```

```
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    Info other = (Info) obj;
    if (contact == null) {
        if (other.contact != null) {
            return false;
        }
    } else if (!contact.equals(other.contact)) {
        return false;
    }
    if (description == null) {
        if (other.description != null) {
            return false;
        }
    } else if (!description.equals(other.description)) {
        return false;
    }
    if (license == null) {
        if (other.license != null) {
             return false;
        }
    } else if (!license.equals(other.license)) {
        return false;
    }
    if (termsOfService == null) {
        if (other.termsOfService != null) {
            return false;
        }
    } else if (!termsOfService.equals(other.termsOfService)) {
        return false;
    }
    if (title == null) {
        if (other.title != null) {
            return false;
        }
    } else if (!title.equals(other.title)) {
        return false;
    }
    if (vendorExtensions == null) {
        if (other.vendorExtensions != null) {
```

```
            return false;
        }
    } else if (!vendorExtensions.equals(other.vendorExtensions)) {
        return false;
    }
    if (version == null) {
        if (other.version != null) {
            return false;
        }
    } else if (!version.equals(other.version)) {
        return false;
     }
    return true;
  }
}
```

Found in path(s):
* /opt/cola/permits/1861058280_1701797798.5796204/0/swagger-models-1-6-2-sources-3-jar/io/swagger/models/Info.java

# 1.156 kotlin-stdlib-common 1.5.31

## 1.156.1 Available under license :

Apache-2.0

# 1.157 kotlinx-metadata-jvm 0.5.0

## 1.157.1 Available under license :

Apache-2.0

# 1.158 woodstox-core 6.2.4

## 1.158.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# 1.159 javax-annotation-api 1.3.2

## 1.159.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8.
Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12.
Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or

otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients. rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient.s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or

such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY.S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item,. as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction.s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts

located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys. fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for
compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California
and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software,
we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to

surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you

also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus

the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute
the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives
a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only
way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear
what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted

interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits
your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular
source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but
you are not obligated to do so.? If you do not wish to do so, delete this exception statement from your version.

# 1.160 isorelax 20090621

## 1.160.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* @(#)$Id: IslandVerifier.java,v 1.6 2003/05/30 23:46:32 kkawa Exp $
*
* Copyright 2001 MURATA Makoto, KAWAGUCHI Kohsuke
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be included
* in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
 * IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
```

* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/2002264690_1714712542.8493288/0/isorelax-20090621-sources-5-jar/org/iso_relax/dispatcher/IslandVerifier.java
No license file was found, but licenses were detected in source scan.

/*
* @(#)$Id: Dispatcher.java,v 1.6 2003/05/30 23:46:32 kkawa Exp $
*
* Copyright 2001 MURATA Makoto, KAWAGUCHI Kohsuke
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be included
* in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
 * IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/2002264690_1714712542.8493288/0/isorelax-20090621-sources-5-jar/org/iso_relax/dispatcher/Dispatcher.java
No license file was found, but licenses were detected in source scan.

/*
* @(#)$Id: IgnoredSchema.java,v 1.5 2003/05/30 23:46:32 kkawa Exp $
*
* Copyright 2001 Kohsuke KAWAGUCHI
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,

* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be included
* in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO
 EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/2002264690_1714712542.8493288/0/isorelax-20090621-sources-5-jar/org/iso_relax/dispatcher/impl/IgnoredSchema.java
No license file was found, but licenses were detected in source scan.

/*
* @(#)$Id: AttributesDecl.java,v 1.3 2001/11/01 00:11:45 kkawa Exp $
*
* Copyright 2001 Kohsuke KAWAGUCHI
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be included
* in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO
 EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/2002264690_1714712542.8493288/0/isorelax-20090621-sources-5-jar/org/iso_relax/dispatcher/AttributesDecl.java
No license file was found, but licenses were detected in source scan.

```
/*
* @(#)$Id: SchemaProvider.java,v 1.6 2003/05/30 23:46:32 kkawa Exp $
*
* Copyright 2001 Kohsuke KAWAGUCHI
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be included
* in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO
 EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/
```

Found in path(s):
* /opt/cola/permits/2002264690_1714712542.8493288/0/isorelax-20090621-sources-5-jar/org/iso_relax/dispatcher/SchemaProvider.java
No license file was found, but licenses were detected in source scan.

```
/*
* @(#)$Id: ElementDecl.java,v 1.3 2001/11/01 00:11:45 kkawa Exp $
*
* Copyright 2001 Kohsuke KAWAGUCHI
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
```

*
* The above copyright notice and this permission notice shall be included
* in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT
 SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/2002264690_1714712542.8493288/0/isorelax-20090621-sources-5-jar/org/iso_relax/dispatcher/ElementDecl.java
No license file was found, but licenses were detected in source scan.

/*
* @(#)$Id: DispatcherImpl.java,v 1.5 2003/05/30 23:46:32 kkawa Exp $
*
* Copyright 2001 Kohsuke KAWAGUCHI
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be included
* in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO
 EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/2002264690_1714712542.8493288/0/isorelax-20090621-sources-5-jar/org/iso_relax/dispatcher/impl/DispatcherImpl.java

No license file was found, but licenses were detected in source scan.

```
/*
* @(#)$Id: IslandSchema.java,v 1.5 2003/05/30 23:46:32 kkawa Exp $
*
* Copyright 2001 KAWAGUCHI Kohsuke
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be included
* in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT
 SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/
```

Found in path(s):
* /opt/cola/permits/2002264690_1714712542.8493288/0/isorelax-20090621-sources-5-jar/org/iso_relax/dispatcher/IslandSchema.java
No license file was found, but licenses were detected in source scan.

```
/*
* @(#)$Id: IslandSchemaReader.java,v 1.4 2001/11/01 00:11:45 kkawa Exp $
*
* Copyright 2001 KAWAGUCHI Kohsuke
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be included
* in all copies or substantial portions of the Software.
```

*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN
 NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/2002264690_1714712542.8493288/0/isorelax-20090621-sources-5-jar/org/iso_relax/dispatcher/IslandSchemaReader.java

# 1.161 jsr305 3.0.2

## 1.161.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2005 Brian Goetz
* Released under the Creative Commons Attribution License
*  (http://creativecommons.org/licenses/by/2.5)
* Official home: http://www.jcip.net
*/

Found in path(s):
* /opt/cola/permits/2035784567_1717163175.5646207/0/jsr305-3-0-2-sources-12-jar/javax/annotation/concurrent/Immutable.java
* /opt/cola/permits/2035784567_1717163175.5646207/0/jsr305-3-0-2-sources-12-jar/javax/annotation/concurrent/GuardedBy.java
* /opt/cola/permits/2035784567_1717163175.5646207/0/jsr305-3-0-2-sources-12-jar/javax/annotation/concurrent/NotThreadSafe.java
* /opt/cola/permits/2035784567_1717163175.5646207/0/jsr305-3-0-2-sources-12-jar/javax/annotation/concurrent/ThreadSafe.java

# 1.162 xml-apis 1.4.01

## 1.162.1 Available under license :

```
 ========================================================================
 ==  NOTICE file corresponding to section 4(d) of the Apache License,   ==
 ==  Version 2.0, in this case for the Apache xml-commons xml-apis      ==
 ==  distribution.                                                      ==
 ========================================================================
```

Apache XML Commons XML APIs
Copyright 1999-2009 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- software copyright (c) 1999, IBM Corporation., http://www.ibm.com.
- software copyright (c) 1999, Sun Microsystems., http://www.sun.com.
- software copyright (c) 2000 World Wide Web Consortium, http://www.w3.org
xml-commons/java/external/LICENSE.sax.txt $Id: LICENSE.sax.txt 225954 2002-01-31 23:26:48Z curcuru $

This license came from: http://www.megginson.com/SAX/copying.html
However please note future versions of SAX may be covered
under http://saxproject.org/?selected=pd

This page is now out of date -- see the new SAX site at
http://www.saxproject.org/ for more up-to-date
releases and other information. Please change your bookmarks.

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for
XML), and release all of the SAX 2.0 source code, compiled code, and
documentation contained in this distribution into the Public Domain.
SAX comes with NO WARRANTY or guarantee of fitness for any
purpose.

David Megginson, david@megginson.com
2000-05-05
xml-commons/java/external/LICENSE.dom-software.txt $Id: LICENSE.dom-software.txt 734314 2009-01-14 03:33:27Z mrglavas $

This license came from: http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/java-binding.zip (COPYRIGHT.html)

W3C SOFTWARE NOTICE AND LICENSE
Copyright 2004 World Wide Web Consortium, (Massachusetts Institute of Technology,
European Research Consortium for Informatics and Mathematics, Keio University).
All Rights Reserved.

The DOM bindings are published under the W3C Software Copyright Notice and
License. The software license requires "Notice of any changes or modifications
to the W3C files, including the date changes were made." Consequently, modified
versions of the DOM bindings must document that they do not conform to the W3C

standard; in the case of the IDL definitions, the pragma prefix can no longer be 'w3c.org'; in the case of the Java language binding, the package names can no longer be in the 'org.w3c' package.

Note: The original version of the W3C Software Copyright Notice and License could be found at http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission.
Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
xml-commons/java/external/LICENSE.dom-documentation.txt $Id: LICENSE.dom-documentation.txt 226215 2005-06-03 22:49:13Z mrglavas $

This license came from: http://www.w3.org/Consortium/Legal/copyright-documents-20021231

W3C DOCUMENT LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231

Public documents on the W3C site are provided by the copyright holders under the following license. By using and/or copying this document, or the W3C document from which this statement is linked, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to copy, and distribute the contents of this document, or the W3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.
2. The pre-existing
copyright notice of the original author, or if it
doesn't exist, a notice (hypertext is preferred, but a textual
representation is permitted) of the form: "Copyright [$date-of-document]
World Wide Web Consortium, (Massachusetts Institute of Technology,
European Research Consortium for Informatics and Mathematics, Keio
University). All Rights Reserved.
http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231"
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software,

documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or derivatives is sometimes granted by the W3C to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

---

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, moves information on style sheets, DTDs, and schemas to the Copyright FAQ, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, such as the translating or annotating specifications. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

Last revised by Reagle $Date: 2005-06-03 18:49:13 -0400 (Fri, 03 Jun 2005) $

# 1.163 aapt2 7.3.0-8691043-osx

## 1.163.1 Available under license :

Copyright (c) 2005-2008, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited
to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.164 layoutlib-api 30.3.0

## 1.164.1 Available under license :

Copyright (c) 2005-2008, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE

file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.165 kotlin-util-io 1.7.10

## 1.165.1 Available under license :

Apache-2.0

# 1.166 kotlin-build-common 1.7.10

## 1.166.1 Available under license :

Apache-2.0

# 1.167 android-test-plugin-result-listener-gradle-proto 30.3.0

## 1.167.1 Available under license :

Copyright (c) 2005-2021, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License

Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated
within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.168 lifecycle-common 2.0.0

## 1.168.1 Available under license :

Apache-2.0

# 1.169 antlr-runtime 4.5.2-1

## 1.169.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
* Copyright (c) 2013 Terence Parr
* Copyright (c) 2013 Sam Harwell
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/pattern/TagChunk.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/pattern/Chunk.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/InterpreterRuleContext.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/pattern/ParseTreePatternMatcher.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/pattern/ParseTreePattern.java

* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/pattern/ParseTreeMatch.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/pattern/TextChunk.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/ParserInterpreter.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/pattern/RuleTagToken.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/pattern/TokenTagToken.java
No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
*  Copyright (c) 2012 Terence Parr
*  Copyright (c) 2012 Sam Harwell
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 THE IMPLIED WARRANTIES
*  OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/PredicateTransition.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/ANTLRFileStream.java

* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/PredictionContextCache.java
*
 /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/InputMismatchException.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/WildcardTransition.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/Pair.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/ParserRuleContext.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ActionTransition.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/TokenFactory.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/PrecedencePredicateTransition.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/ParseTreeListener.java
*
 /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/Token.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/PredictionMode.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/AbstractPredicateTransition.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/TokenSource.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/NoViableAltException.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/BufferedTokenStream.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/ParseCancellationException.java
*
 /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/Tree.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/UnbufferedCharStream.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ATNSerializer.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/AbstractParseTreeVisitor.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/BaseErrorListener.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/TokenStreamRewriter.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-

jar/org/antlr/v4/runtime/misc/AbstractEqualityComparator.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/BailErrorStrategy.java
*
/opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/StarLoopbackState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/dfa/DFA.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/Interval.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/Triple.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/RuleTransition.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/Recognizer.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ATNConfig.java
*
/opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/dfa/DFAState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/LexerInterpreter.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/PredictionContext.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/CommonToken.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/ConsoleErrorListener.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/RangeTransition.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/IntegerStack.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/Array2DHashSet.java
*
/opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/ParseTreeWalker.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/dfa/DFASerializer.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/ErrorNodeImpl.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/SingletonPredictionContext.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerATNSimulator.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/TerminalNodeImpl.java

* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/RecognitionException.java
*
 /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/PlusBlockStartState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ArrayPredictionContext.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LoopEndState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/RuleNode.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/RuleContext.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/IntSet.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/EpsilonTransition.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/ANTLRErrorStrategy.java
*
 /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/BlockStartState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/Parser.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/SyntaxTree.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/Utils.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/ProxyErrorListener.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/OrderedHashSet.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/TerminalNode.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/LogManager.java
*
 /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/dfa/LexerDFASerializer.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/StarLoopEntryState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/PlusLoopbackState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/MurmurHash.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/Trees.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-

jar/org/antlr/v4/runtime/DefaultErrorStrategy.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/OrderedATNConfigSet.java
*
/opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/RuleStartState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/SetTransition.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ATN.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerATNConfig.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ATNConfigSet.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/MultiMap.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/ParseTree.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/FailedPredicateException.java
*
/opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/WritableToken.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/SemanticContext.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LL1Analyzer.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ParserATNSimulator.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/UnbufferedTokenStream.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/IntStream.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/CommonTokenStream.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/ParseTreeProperty.java
*
/opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/DiagnosticErrorListener.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/ANTLRErrorListener.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/ObjectEqualityComparator.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/TokensStartState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/CommonTokenFactory.java

* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/DoubleKeyMap.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/IntegerList.java
*
 /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/DecisionState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/BasicState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/Transition.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/CharStream.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/ParseTreeVisitor.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/IntervalSet.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/TokenStream.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/BasicBlockStartState.java
*
 /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/tree/ErrorNode.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/Lexer.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/EqualityComparator.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/StarBlockStartState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/RuleStopState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/NotSetTransition.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/LexerNoViableAltException.java
*
 /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/ANTLRInputStream.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/AtomTransition.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/misc/FlexibleHashMap.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ATNState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/BlockEndState.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-

jar/org/antlr/v4/runtime/misc/NotNull.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/EmptyPredictionContext.java
No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
*  Copyright (c) 2013 Terence Parr
*  Copyright (c) 2013 Sam Harwell
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 THE IMPLIED WARRANTIES
*  OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerModeAction.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerMoreAction.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerChannelAction.java
*
 /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerAction.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ATNSimulator.java

* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerActionType.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ATNDeserializationOptions.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerPopModeAction.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerIndexedCustomAction.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ATNDeserializer.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerSkipAction.java
*
/opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ATNType.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ProfilingATNSimulator.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerActionExecutor.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerPushModeAction.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerTypeAction.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LexerCustomAction.java
No license file was found, but licenses were detected in source scan.

/*
* [The "BSD license"]
*  Copyright (c) 2014 Terence Parr
*  Copyright (c) 2014 Sam Harwell
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,

THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/AmbiguityInfo.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/LookaheadEventInfo.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ContextSensitivityInfo.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/Vocabulary.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/DecisionEventInfo.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/DecisionInfo.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/VocabularyImpl.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ErrorInfo.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/ParseInfo.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/atn/PredicateEvalInfo.java
* /opt/cola/permits/2100345344_1721125029.8350801/0/antlr4-runtime-4-5-2-1-sources-jar/org/antlr/v4/runtime/RuntimeMetaData.java

# 1.170 lint-typedef-remover 30.3.0

## 1.170.1 Available under license :

Copyright (c) 2011, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to
compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within
the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License,

without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible
for determining the
appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.171 kotlin-klib-commonizer-api 1.7.10

## 1.171.1 Available under license :

Apache-2.0

# 1.172 kotlin-scripting-compiler-impl-embeddable 1.5.31

## 1.172.1 Available under license :

Apache-2.0

# 1.173 javawriter 2.5.0

## 1.173.1 Available under license :

Apache-2.0

# 1.174 common 30.3.0

## 1.174.1 Available under license :

Copyright (c) 2005-2013, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely

responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.175 dagger 2.44

## 1.175.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2015 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/multibindings/LongKey.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/multibindings/Multibinds.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/multibindings/StringKey.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/Subcomponent.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/multibindings/ClassKey.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/multibindings/IntKey.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/Beta.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/DaggerCollections.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/MapProviderFactory.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/SetFactory.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/MembersInjectors.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/SingleCheck.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/InstanceFactory.java

* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/MapKey.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/AbstractMapFactory.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/MapFactory.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/DelegateFactory.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/Component.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/Factory.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/assisted/AssistedInject.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/assisted/Assisted.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/assisted/AssistedFactory.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2019 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
```

*/

Found in path(s):
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/InjectedFieldSignature.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/ComponentDefinitionType.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/MapBuilder.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/SetBuilder.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2022 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-

jar/dagger/internal/ScopeMetadata.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/QualifierMetadata.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/Provides.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/GwtIncompatible.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/ProviderOfLazy.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/Preconditions.java
*

/opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/multibindings/ElementsIntoSet.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/assisted/package-info.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/Reusable.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/multibindings/IntoMap.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/BindsInstance.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/multibindings/package-info.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/BindsOptionalOf.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/multibindings/IntoSet.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/DoubleCheck.java
*
/opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/Binds.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/Lazy.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/MembersInjector.java
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/Module.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345257_1721125104.9500396/0/dagger-2-44-sources-jar/dagger/internal/DaggerGenerated.java

# 1.176 kotlinx-coroutines-android 1.5.0

## 1.176.1 Available under license :

Apache-2.0

# 1.177 aaptcompiler 7.3.0

## 1.177.1 Available under license :

Copyright (c) 2005-2008, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited
to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.178 openapi-generator-gradle-plugin 5.2.1

## 1.178.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2018 OpenAPI-Generator Contributors (https://openapi-generator.tech)
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/2100344712_1721124839.4115787/0/openapi-generator-gradle-plugin-5-2-1-sources-jar/org/openapitools/generator/gradle/plugin/tasks/GeneratorsTask.kt
* /opt/cola/permits/2100344712_1721124839.4115787/0/openapi-generator-gradle-plugin-5-2-1-sources-jar/org/openapitools/generator/gradle/plugin/extensions/OpenApiGeneratorGenerateExtension.kt
*
 /opt/cola/permits/2100344712_1721124839.4115787/0/openapi-generator-gradle-plugin-5-2-1-sources-jar/org/openapitools/generator/gradle/plugin/tasks/ValidateTask.kt
* /opt/cola/permits/2100344712_1721124839.4115787/0/openapi-generator-gradle-plugin-5-2-1-sources-jar/org/openapitools/generator/gradle/plugin/extensions/OpenApiGeneratorValidateExtension.kt
* /opt/cola/permits/2100344712_1721124839.4115787/0/openapi-generator-gradle-plugin-5-2-1-sources-jar/org/openapitools/generator/gradle/plugin/OpenApiGeneratorPlugin.kt
* /opt/cola/permits/2100344712_1721124839.4115787/0/openapi-generator-gradle-plugin-5-2-1-sources-jar/org/openapitools/generator/gradle/plugin/tasks/GenerateTask.kt
* /opt/cola/permits/2100344712_1721124839.4115787/0/openapi-generator-gradle-plugin-5-2-1-sources-jar/org/openapitools/generator/gradle/plugin/extensions/OpenApiGeneratorMetaExtension.kt
* /opt/cola/permits/2100344712_1721124839.4115787/0/openapi-generator-gradle-plugin-5-2-1-sources-jar/org/openapitools/generator/gradle/plugin/tasks/MetaTask.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2019 OpenAPI-Generator Contributors (https://openapi-generator.tech)
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344712_1721124839.4115787/0/openapi-generator-gradle-plugin-5-2-1-sources-jar/org/openapitools/generator/gradle/plugin/extensions/OpenApiGeneratorGeneratorsExtension.kt

# 1.179 semver4j 0.16.4

## 1.179.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 David Gundersen <gundy@acm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

# 1.180 kotlin-compiler-embeddable 1.7.10

## 1.180.1 Available under license :

Copyright (C) 2002-2020 Sebastiano Vigna

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.181 lifecycle-common 2.3.1

## 1.181.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at

*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/GenericLifecycleObserver.java
* /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/Lifecycle.java
* /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/CompositeGeneratedAdaptersObserver.java
*
 /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/FullLifecycleObserver.java
* /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/LifecycleObserver.java
* /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/LifecycleOwner.java
* /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/GeneratedAdapter.java
* /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/ClassesInfoCache.java
* /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/Lifecycling.java
* /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/MethodCallsLogger.java
*
 /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/SingleGeneratedAdapterObserver.java
* /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/ReflectiveGenericLifecycleObserver.java
* /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/OnLifecycleEvent.java
* /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/FullLifecycleObserverAdapter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2018 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345005_1721146619.1743243/0/lifecycle-common-2-3-1-sources-jar/androidx/lifecycle/LifecycleEventObserver.java

# 1.182 kotlin-compiler-runner 1.7.10

## 1.182.1 Available under license :

Apache-2.0

# 1.183 kotlin-annotation-processing-gradle 1.7.10

## 1.183.1 Available under license :

Apache-2.0

# 1.184 kotlin-gradle-plugin 1.7.10-gradle70

## 1.184.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2015 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and

* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/utils/Printer.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/dsl/KotlinMultiplatformCommonOptionsImpl.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/dsl/KotlinJsDceOptionsImpl.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/tasks/KotlinJsDce.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/internal/subpluginUtils.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/internal/CompilerArgumentAware.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/internal/CompilerArgumentsGradleInput.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/tasks/CacheableTasks.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/compilerRunner/reportUtils.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/dsl/optionsUtil.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/internal/kapt/KaptGenerateStubsTask.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-

jar/org/jetbrains/kotlin/gradle/utils/compatibiltiy.kt
No license file was found, but licenses were detected in source scan.

var license: String? = null

Found in path(s):
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/targets/native/cocoapods/CocoapodsExtension.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2016 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/dsl/KotlinJsOptionsImpl.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/utils/ParsedGradleVersion.kt
*
 /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/plugin/gradleUtils.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/plugin/KaptExtension.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/tasks/CompilerPluginOptions.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/dsl/KotlinCompile.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/tasks/KotlinCompileCommon.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/plugin/KotlinPluginWrapper.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/tasks/TasksProvider.kt
*

/opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/dsl/KotlinJvmOptionsImpl.kt
* /opt/cola/permits/2100345365_1721144506.857916/0/kotlin-gradle-plugin-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/plugin/KotlinMultiplatformPlugin.kt

# 1.185 kotlin-native-utils 1.7.10

## 1.185.1 Available under license :

Apache-2.0

# 1.186 lint-model 30.3.0

## 1.186.1 Available under license :

Copyright (c) 2011, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to
compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within

the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License,
without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible
for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.187 kotlin-native-utils 1.5.31

## 1.187.1 Available under license :

Apache-2.0

# 1.188 repository 30.3.0

## 1.188.1 Available under license :

Copyright (c) 2011, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within
the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.189 transform-api 2.0.0-deprecated-use-gradle-api

### 1.189.1 Available under license :

Apache-2.0

## 1.190 kotlin-daemon-client 1.7.10

### 1.190.1 Available under license :

Apache-2.0

## 1.191 gradle-api 7.3.0

### 1.191.1 Available under license :

Copyright (c) 2005-2017, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form
of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.192 kotlin-android-extensions 1.5.31

## 1.192.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2015 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/parcel/ParcelableClinitClassBuilderInterceptorExtension.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/diagnostic/AndroidExtensionPropertiesCallChecker.kt
*

/opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/res/AndroidPackageFragmentProviderExtension.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/res/CliAndroidLayoutXmlFileManager.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/res/androidResources.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/descriptors/LazySyntheticElementResolveContext.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/AndroidComponentRegistrar.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/res/AndroidLayoutXmlFileManager.kt
*
/opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/codegen/AndroidOnDestroyClassBuilderInterceptorExtension.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/diagnostic/DefaultErrorMessagesAndroid.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/res/CliAndroidPackageFragmentProviderExtension.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/AndroidConst.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/diagnostic/ErrorsAndroid.java
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/AndroidXmlHandler.kt
*
/opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/descriptors/AndroidSyntheticPackageFragmentDescriptor.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/descriptors/PredefinedPackageFragmentDescriptor.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2016 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/res/syntheticDescriptorGeneration.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/kotlinx/android/parcel/RawValue.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/kotlinx/android/parcel/Parcelize.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/parcel/serializers/ParcelSerializers.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/parcel/serializers/ParcelSerializer.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/codegen/AndroidContainerType.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/codegen/CacheMechanisms.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/kotlinx/android/parcel/Parceler.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/kotlinx/android/parcel/WriteWith.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/descriptors/ContainerOptionsProxy.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/kotlinx/android/extensions/LayoutContainer.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/kotlinx/android/extensions/CacheImplementation.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/codegen/CliAndroidOnDestroyClassBuilderInterceptorExtension.kt

* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/kotlinx/android/extensions/ContainerOptions.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/synthetic/codegen/CliAndroidExtensionsExpressionCodegenExtension.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/kotlinx/android/parcel/TypeParceler.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/parcel/ParcelableResolveExtension.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/org/jetbrains/kotlin/android/parcel/ParcelableAnnotationChecker.kt
* /opt/cola/permits/2100344647_1721124963.84039/0/kotlin-android-extensions-1-5-31-sources-jar/kotlinx/android/parcel/IgnoredOnParcel.kt

# 1.193 kotlin-annotation-processing-gradle 1.5.31

## 1.193.1 Available under license :

Apache-2.0

# 1.194 builder-model 7.3.0

## 1.194.1 Available under license :

Copyright (c) 2005-2013, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.195 dagger-compiler 2.44

## 1.195.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2013 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
```

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/ElementFormatter.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/Util.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/AssistedInjectProcessingStep.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/UnscopedDirectInstanceRequestRepresentationFactory.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/ValidationBindingGraphPlugins.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/SwitchingProviderInstanceSupplier.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/ServiceLoaders.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/DirectInstanceBindingRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/UnscopedFrameworkInstanceCreationExpressionFactory.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/StaticMemberSelects.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-

jar/dagger/internal/codegen/processingstep/SuperficialValidator.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/writing/FrameworkInstanceBindingRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/writing/BindingRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/writing/ProviderInstanceSupplier.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/writing/StaticFactoryInstanceSupplier.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/base/TarjanSCCs.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/writing/BindingRepresentations.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/bindinggraphvalidation/SetMultibindingValidator.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/writing/MembersInjectionBindingRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/writing/ProvisionBindingRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/validation/ExternalBindingGraphPlugins.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/writing/ExperimentalSwitchingProviderDependencyRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/validation/ExternalBindingGraphConverter.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/base/DaggerSuperficialValidation.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/writing/ProductionBindingRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-
jar/dagger/internal/codegen/base/ProducerAnnotations.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/TypeCheckingProcessingStep.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/bindinggraphvalidation/MapMultibindingValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/Validation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/bindinggraphvalidation/BindingGraphValidationModule.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/OptionalFactoryInstanceCreationExpression.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/SwitchingProviders.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/compileroption/ProcessingOptions.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/InjectBindingRegistryModule.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/SourceFileGeneratorsModule.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/DiagnosticMessageGenerator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/BindingRequest.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/DependencyRequestValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/InaccessibleMapKeyProxyGenerator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/bindinggraphvalidation/DependsOnProductionExecutorValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/DiagnosticReporterFactory.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/SubcomponentCreatorBindingEdgeImpl.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/DependencyEdgeImpl.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/MembersInjectionValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/ProcessingEnvironmentModule.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/BindingMethodValidatorsModule.java

* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/ComponentNodeImpl.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/PerComponentImplementation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/bindinggraphvalidation/ProvisionDependencyOnProducerBindingValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/ChildFactoryMethodEdgeImpl.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/bindinggraphvalidation/SubcomponentFactoryMethodValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/PerGeneratedFile.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/InstanceFactoryCreationExpression.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/componentgenerator/TopLevelImplementationComponent.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ParentComponent.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/bindinggraphvalidation/MissingBindingValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ProviderInstanceRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/BindingGraphValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ProducerNodeInstanceRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ProducerEntryPointView.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/BindingGraphConverter.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ModuleGenerator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/DerivedFromFrameworkInstanceRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/BindingNode.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/MethodRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/bindinggraphvalidation/NullableBindingValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ImmediateFutureRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/bindinggraphvalidation/DuplicateBindingsValidator.java
*

/opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ModuleProxies.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/MultibindingFactoryCreationExpression.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/bindinggraphvalidation/InjectBindingValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/componentgenerator/CurrentImplementationSubcomponent.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/ComponentDescriptorValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/bindinggraphvalidation/IncompatiblyScopedBindingsValidator.java
*
/opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/bindinggraphvalidation/DependencyCycleValidator.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/FrameworkTypes.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/SetType.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/ComponentCreatorValidator.java
*
/opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/MapKeys.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/ContributionType.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/ResolvedBindings.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-

jar/dagger/internal/codegen/writing/TopLevel.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/ModuleKind.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/MonitoringModuleProcessingStep.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/MapType.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/InjectionOrProvisionProviderCreationExpression.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/ModuleDescriptor.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ProducerCreationExpression.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/DependencyMethodProviderCreationExpression.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/BindingType.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/DelegatingFrameworkInstanceCreationExpression.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/MapFactoryCreationExpression.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/MemberSelect.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ComponentNames.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/MultibindingDeclaration.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/langmodel/Accessibility.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/MembersInjectorProviderCreationExpression.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/ComponentHierarchyValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/BindingDeclarationFormatter.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/BindingDeclaration.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ProducerFromProviderCreationExpression.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/MonitoringModuleGenerator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/DependencyMethodProducerCreationExpression.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/SetFactoryCreationExpression.java

* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/AnonymousProviderCreationExpression.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/FrameworkFieldInitializer.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/compileroption/ValidationType.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/MultibindsMethodValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/SubcomponentDeclaration.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/BindsInstanceMethodValidator.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/BindsInstanceProcessingStep.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/BindsMethodValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/BindsInstanceElementValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/javapoet/AnnotationSpecs.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ComponentRequestRepresentations.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/BindingElementValidator.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/AnyBindingMethodValidator.java

* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/OptionalFactories.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/DelegateDeclaration.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/BindingMethodValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/AnnotationExpression.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/javapoet/TypeSpecs.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/javapoet/CodeBlocks.java
*
/opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/package-info.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/ComponentRequirement.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/FrameworkType.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/javapoet/TypeNames.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/OptionalType.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/BindsOptionalOfMethodValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/compileroption/FeatureStatus.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/MultibindingAnnotationsProcessingStep.java
*
/opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/UnwrappedMapKeyGenerator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/OptionalBindingDeclaration.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/UniqueNameSet.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ComponentImplementation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/compileroption/CompilerOptions.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/BindsInstanceParameterValidator.java
*
/opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/BindingMethodProcessingStep.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/SimpleMethodRequestRepresentation.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2019 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/compileroption/ProcessingEnvironmentCompilerOptions.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/ProcessingRoundCacheModule.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/ClearableCache.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/ComponentCreatorKind.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/kotlin/KotlinMetadataFactory.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/ComponentAnnotation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/kotlin/KotlinMetadata.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/ComponentCreatorAnnotation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/bindinggraphvalidation/CompositeBindingGraphPlugin.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/kotlin/KotlinMetadataUtil.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/ModuleAnnotation.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2017 The Dagger Authors.
*
```

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/HjarSourceFileGenerator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/DelegateRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/OptionalRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/RequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/MapKeyAccessibility.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/Keys.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ComponentRequirementExpressions.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/DiagnosticFormatting.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/Scopes.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/InjectionSiteFactory.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/InjectionMethods.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/SubcomponentCreatorRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/GwtCompatibility.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ComponentMethodRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/ComponentHjarProcessingStep.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/BindsTypeChecker.java

* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ComponentRequirementRequestRepresentation.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/FrameworkInstanceRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ComponentCreatorImplementationFactory.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/MembersInjectionMethods.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/PrivateMethodRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/AssistedInjectionParameters.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ComponentRequirementExpression.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ComponentCreatorImplementation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/MapRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/javapoet/Expression.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/InjectBindingRegistry.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ComponentInstanceRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/FrameworkInstanceSupplier.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/SetRequestRepresentation.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/MembersInjectionRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ComponentProvisionRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/BindingFactory.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/BindingGraphFactory.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/MethodSignatureFormatter.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/ModuleValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/FrameworkField.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/SourceFiles.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/ComponentProcessor.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/FrameworkTypeMapper.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/ComponentDescriptorFactory.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/DependencyRequestFactory.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/MembersInjectorGenerator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/ConfigurationAnnotations.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/FactoryGenerator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/AnnotationCreatorGenerator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/ComponentCreatorDescriptor.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/LegacyBindingGraph.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/componentgenerator/ComponentGenerator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/ComponentDescriptor.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/ProducesMethodValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-

jar/dagger/internal/codegen/validation/ComponentValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/SourceFileGenerator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/MethodSignature.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/KeyVariableNamer.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ProducerFactoryGenerator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/DependencyVariableNamer.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/ModuleProcessingStep.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/SourceFileGenerationException.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/MapKeyValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/InjectValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/InjectionAnnotations.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/Binding.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/MapKeyProcessingStep.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/ComponentProcessingStep.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/ProductionBinding.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/Formatter.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/MembersInjectionBinding.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/ContributionBinding.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/DelegateComponentProcessor.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/ErrorMessages.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/DependencyRequestFormatter.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/ComponentKind.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/KeyFactory.java

* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/BindingGraph.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/InjectBindingRegistryImpl.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/ValidationReport.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/InjectProcessingStep.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/ProvisionBinding.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/validation/ProvidesMethodValidator.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/base/RequestKinds.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/binding/AssistedInjectionAnnotations.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/AssistedFactoryProcessingStep.java
*
 /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/AssistedProcessingStep.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/bindinggraphvalidation/PackageNameCompressor.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/componentgenerator/ComponentGeneratorModule.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/AssistedFactoryRequestRepresentation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-

jar/dagger/internal/codegen/componentgenerator/ComponentHjarGenerator.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2022 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/processingstep/ProcessingStepsModule.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/KspComponentProcessor.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/FrameworkInstanceKind.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/GeneratedImplementation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ComponentWrapperImplementation.java
* /opt/cola/permits/2100344842_1721124972.817544/0/dagger-compiler-2-44-sources-jar/dagger/internal/codegen/writing/ExperimentalSwitchingProviders.java

# 1.196 kotlin-stdlib-jdk7 1.5.0

## 1.196.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
```

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345246_1721124789.601679/0/kotlin-stdlib-jdk7-1-5-0-sources-jar/kotlin/AutoCloseable.kt

# 1.197 android-test-plugin-host-additional-test-output-proto 30.3.0

## 1.197.1 Available under license :

Copyright (c) 2005-2021, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited
to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated

within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.198 apkzlib 7.3.0

## 1.198.1 Available under license :

Copyright (c) 2005-2013, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form

of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.199 kotlin-tooling-metadata 1.5.31

## 1.199.1 Available under license :

Apache-2.0

# 1.200 bundletool 1.9.0

## 1.200.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.201 kotlin-stdlib 1.8.0

## 1.201.1 Available under license :

Apache-2.0

# 1.202 kotlin-project-model 1.7.10

## 1.202.1 Available under license :

Apache-2.0

# 1.203 kotlinx-coroutines-android 1.6.1

## 1.203.1 Available under license :

Apache-2.0

# 1.204 jetifier-processor 1.0.0-beta10

## 1.204.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2019 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
```

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/AndroidXRefScanner.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/SignatureDetection.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/bytecode/CoreRemapperImpl.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/proguard/patterns/ReplacersRunner.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/pom/PomDocument.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/metainf/MetaInfTransformer.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/proguard/ProGuardClassSpecParser.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/proguard/ProGuardTransformer.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/proguard/patterns/PatternHelper.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/pom/PomScanner.kt

*
/opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/bytecode/CoreRemapper.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/bytecode/ByteCodeTransformer.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/proguard/ProGuardClassFilterParser.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/proguard/patterns/GroupsReplacer.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/proguard/ProGuardTypesMapper.kt
*
/opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/pom/XmlUtils.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/bytecode/asm/CustomRemapper.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2018 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/AnnotationFilesMerger.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/archive/ArchiveFile.kt
*
/opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/CollectionExtensions.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/archive/ArchiveItemVisitor.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/resource/XmlResourcesTransformer.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/FileMapping.kt

* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/SourceJetifier.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/archive/Archive.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/Transformer.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/TransformationResult.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/pom/PomDependencyExtensions.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/archive/ArchiveItem.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/Processor.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/type/MapGeneratorRemapper.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/transform/TransformationContext.kt
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/type/LibraryMapGenerator.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2020 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345460_1721147866.2852912/0/jetifier-processor-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/processor/TimestampsPolicy.kt

# 1.205 annotation 1.2.0

## 1.205.1 Available under license :

Apache Olingo
Copyright 2013-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.206 kotlin-stdlib-jdk7 1.8.0

## 1.206.1 Available under license :

Apache-2.0

# 1.207 kotlin-stdlib-jdk8 1.8.0

## 1.207.1 Available under license :

Apache-2.0

# 1.208 databinding-compiler-common 7.3.0

## 1.208.1 Available under license :

Copyright (c) 2005-2017, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited
to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.209 databinding-common 7.3.0

## 1.209.1 Available under license :

Copyright (c) 2005-2017, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.210 kotlin-gradle-plugin-model 1.7.10

## 1.210.1 Available under license :

Apache-2.0

# 1.211 symbol-processing-api 1.7.0-1.0.6

## 1.211.1 Available under license :

Apache-2.0

# 1.212 tensorflow-lite-metadata 0.1.0-rc2

## 1.212.1 Available under license :

Apache-2.0

# 1.213 kotlin-build-common 1.5.31

## 1.213.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2015 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
```

* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/IdToFileMap.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/LocalFileKotlinClass.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/compilerRunner/OutputItemsCollectorImpl.java
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/build/JvmSourceRoot.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/java_descriptors.proto
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/FileToIdMap.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/compilerRunner/OutputItemsCollector.java
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/CachingLazyStorage.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/BasicMap.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/modules/KotlinModuleXmlBuilder.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/LookupMap.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/BasicMapsOwner.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/build/generatedFiles.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/ClassAttributesMap.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/compilerRunner/SimpleOutputItem.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/LazyStorage.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/LookupStorage.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/IncrementalCompilationComponentsImpl.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/ClassOneToManyMap.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/NonCachingLazyStorage.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-

jar/org/jetbrains/kotlin/incremental/storage/values.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/externalizers.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2016 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/protoDifferenceUtils.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/fileUtils.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/IncrementalJvmCache.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/buildUtil.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/compilerRunner/ArgumentUtils.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
```

* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/build/serializationUtils.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/build/JvmBuildMetaInfo.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/SourceToOutputMaps.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/DirtyClassesMaps.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/ChangesCollector.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/compilerRunner/MessageCollectorToOutputItemsCollectorAdapter.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/AbstractIncrementalCache.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/storage/NameTransformers.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/IncrementalJsCache.kt
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/JavaClassesSerializerExtension.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345429_1721125143.9181943/0/kotlin-build-common-1-5-31-sources-jar/org/jetbrains/kotlin/incremental/ProtoCompareGenerated.kt

# 1.214 kotlin-gradle-plugin-model 1.5.31

## 1.214.1 Available under license :

Apache-2.0

# 1.215 dagger-producers 2.44

## 1.215.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/ProductionImplementation.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/AbstractMapProducer.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/MapOfProducerProducer.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/ProductionScope.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/ProductionSubcomponent.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/Production.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/MapProducer.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/package-info.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-

jar/dagger/producers/internal/MapOfProducedProducer.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2021 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/AnnotationUsages.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2014 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/Producers.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/Producer.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/Producers.java
*

/opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/Produced.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/Produces.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/ProductionComponent.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/ProducerModule.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2015 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/monitoring/ProductionComponentMonitor.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/monitoring/TimingProductionComponentMonitor.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/monitoring/package-info.java
*
/opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/monitoring/ProducerMonitor.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/SetProducer.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/monitoring/ProducerToken.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/monitoring/TimingProducerMonitor.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/SetOfProducedProducer.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/monitoring/internal/Monitors.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/monitoring/ProducerTimingRecorder.java

*
/opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/monitoring/ProductionComponentTimingRecorder.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/monitoring/TimingRecorders.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/AbstractProducer.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/CancellationListener.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/AbstractProducesMethodProducer.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/CancellableProducer.java
*
/opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/CancellationPolicy.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/DependencyMethodProducer.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/DelegateProducer.java
* /opt/cola/permits/2100345277_1721125160.6909113/0/dagger-producers-2-44-sources-jar/dagger/producers/internal/ProductionExecutorModule.java

# 1.216 aapt2-proto 7.3.0-8691043

## 1.216.1 Available under license :

Copyright (c) 2005-2008, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited
to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.217 signflinger 7.3.0

## 1.217.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2019 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345439_1721147857.9901652/0/signflinger-7-3-0-sources-jar/com/android/signflinger/SignedApkOptions.java
* /opt/cola/permits/2100345439_1721147857.9901652/0/signflinger-7-3-0-sources-jar/com/android/signflinger/ApkSigningBlock.java
* /opt/cola/permits/2100345439_1721147857.9901652/0/signflinger-7-3-0-sources-jar/com/android/signflinger/SignedApk.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
```

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345439_1721147857.9901652/0/signflinger-7-3-0-sources-jar/com/android/signflinger/SigningBlockUtils.java

# 1.218 kotlin-klib-commonizer-api 1.5.31

## 1.218.1 Available under license :

Apache-2.0

# 1.219 kotlin-stdlib-jdk8 1.7.0

## 1.219.1 Available under license :

Apache-2.0

# 1.220 tink 1.3.0-rc2

## 1.220.1 Available under license :

No license file was found, but licenses were detected in source scan.

// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//  http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,
/**
* Poly1305 one-time MAC based on RFC 7539.
*
* <p>This is not an implementation of the MAC interface on purpose and it is not equivalent to
* HMAC.
*
* <p>The implementation is based on poly1305 implementation by Andrew Moon
* (https://github.com/floodyberry/poly1305-donna) and released as public domain.
*/

Found in path(s):
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Poly1305.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2010 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Base64.java
No license file was found, but licenses were detected in source scan.

// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//  http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/ChaCha20.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/EciesAeadHkdfHybridEncrypt.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/SignatureKeyTemplates.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/AesCmac.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/hybrid/EciesAeadHkdfPublicKeyManager.java
*
 /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/EcdsaSignJce.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/EngineWrapper.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/mac/MacKeyTemplates.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/streamingaead/StreamingAeadWrapper.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/hybrid/HybridConfig.java

* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/daead/DeterministicAeadConfig.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/PublicKeySignConfig.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/StreamingAeadDecryptingChannel.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Ed25519Sign.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/ChaCha20Poly1305Base.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/config/TinkConfig.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Hex.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/aead/ChaCha20Poly1305KeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/streamingaead/AesGcmHkdfStreamingKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/mac/MacConfig.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/XChaCha20Poly1305.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/hybrid/EciesAeadHkdfPrivateKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Random.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/StreamingAeadEncryptingChannel.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/aead/AeadConfig.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/aead/AeadKeyTemplates.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/RsaSsaPssVerifyJce.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/PublicKeyVerifyWrapper.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/mac/MacFactory.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/SignaturePemKeysetReader.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/aead/KmsAeadKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/aead/AeadFactory.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-

jar/com/google/crypto/tink/streamingaead/AesCtrHmacStreamingKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/streamingaead/StreamingAeadHelper.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/subtle/ChaCha20Base.java
*
/opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/subtle/AesUtil.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/mac/AesCmacKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/subtle/EcdsaVerifyJce.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/subtle/EciesAeadHkdfDemHelper.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/subtle/MacJce.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/signature/EcdsaSignKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/streamingaead/StreamingAeadFactory.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/daead/AesSivKeyManager.java
*
/opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/subtle/Ed25519Verify.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/subtle/ImmutableByteArray.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/HybridEncrypt.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/PublicKeyVerify.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/streamingaead/StreamingAeadKeyTemplates.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/subtle/StreamSegmentEncrypter.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/subtle/EciesHkdfRecipientKem.java
*
/opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/subtle/StreamingAeadEncryptingStream.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/hybrid/HybridEncryptFactory.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/signature/PublicKeySignWrapper.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/hybrid/RegistryEciesAeadHkdfDemHelper.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-
jar/com/google/crypto/tink/signature/PublicKeyVerifyConfig.java

* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/EngineFactory.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/Mac.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/streamingaead/StreamingAeadUtil.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/EciesAeadHkdfHybridDecrypt.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Hkdf.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/NonceBasedStreamingAead.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/daead/DeterministicAeadWrapper.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/SigUtil.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/StreamingAeadDecryptingStream.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/streamingaead/StreamingAeadConfig.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/AesSiv.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/RsaSsaPkcs1VerifyJce.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/X25519.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/mac/MacWrapper.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/EcdsaVerifyKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/daead/DeterministicAeadKeyTemplates.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/StreamSegmentDecrypter.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/Ed25519PrivateKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/RewindableReadableByteChannel.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/PublicKeySign.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Kwp.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/aead/AeadWrapper.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-

jar/com/google/crypto/tink/StreamingAead.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Ed25519Constants.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/daead/DeterministicAeadFactory.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/mac/HmacKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/hybrid/HybridDecryptConfig.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/RsaSsaPssVerifyKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Field25519.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/aead/AesCtrHmacAeadKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/IndCpaCipher.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/aead/KmsEnvelopeAead.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/hybrid/HybridEncryptConfig.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/CleartextKeysetHandle.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/aead/XChaCha20Poly1305KeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Ed25519.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/EllipticCurves.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/HybridDecrypt.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/hybrid/HybridKeyTemplates.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/AesGcmHkdfStreaming.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/SignatureConfig.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/aead/KmsEnvelopeAeadKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/hybrid/HybridDecryptFactory.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Validators.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Bytes.java

* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/streamingaead/ReadableByteChannelDecrypter.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/KeyWrap.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/hybrid/HybridEncryptWrapper.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/PublicKeySignFactory.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/AesCtrHmacStreaming.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/PemKeyType.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/RsaSsaPkcs1VerifyKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/aead/AesEaxKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/EciesHkdfSenderKem.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/XChaCha20.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Enums.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/RsaSsaPssSignJce.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/hybrid/HybridUtil.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/ChaCha20Poly1305.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/EncryptThenAuthenticate.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/Curve25519.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/streamingaead/SeekableByteChannelDecrypter.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/SubtleUtil.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/RsaSsaPssSignKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/streamingaead/InputStreamDecrypter.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/hybrid/HybridDecryptWrapper.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/RsaSsaPkcs1SignKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-

jar/com/google/crypto/tink/aead/AesGcmKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/Ed25519PublicKeyManager.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/RsaSsaPkcs1SignJce.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/util/KeysDownloader.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/Aead.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/signature/PublicKeyVerifyFactory.java
No license file was found, but licenses were detected in source scan.

// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//  http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,
// unique. If one single IV is ever used to encrypt two or more messages, the confidentiality of

Found in path(s):
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/AesCtrJceCipher.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/aead/AesCtrKeyManager.java
No license file was found, but licenses were detected in source scan.

// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//  http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,
/**
 * A description of the state of this StreamingAeadSeekableDecryptingChannel.
 * While this description does not contain plaintext or key material
 * it contains length information that might be confidential.
 */

Found in path(s):
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/StreamingAeadSeekableDecryptingChannel.java
No license file was found, but licenses were detected in source scan.

// // Licensed under the Apache License, Version 2.0 (the "License");
// // you may not use this file except in compliance with the License.
// // You may obtain a copy of the License at
// //  http://www.apache.org/licenses/LICENSE-2.0

// // distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/AesGcmJce.java
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/subtle/AesEaxJce.java
No license file was found, but licenses were detected in source scan.

```
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//      http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,
/**
  * Deterministically encrypts {@code plaintext} with {@code associatedData} as associated
  * authenticated data.
  *
  * <p><b>Warning</b>
  *
  * <p>Encrypting the same {@code plaintext} multiple times protects the integrity of that
  * plaintext, but confidentiality is compromised to the extent that an attacker can determine that
  * the same plaintext was encrypted.
  *
  * <p>The resulting ciphertext allows for checking authenticity and integrity of associated data
  * ({@code associatedData}), but does not guarantee its secrecy.
  *
  * @return resulting ciphertext
  */
```

Found in path(s):
* /opt/cola/permits/2100345067_1721124784.3612206/0/tink-1-3-0-rc2-sources-jar/com/google/crypto/tink/DeterministicAead.java

# 1.221 kotlin-sam-with-receiver 1.5.31

## 1.221.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
```

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345168_1721125089.8386343/0/kotlin-sam-with-receiver-1-5-31-sources-jar/org/jetbrains/kotlin/noarg/gradle/SamWithReceiverExtension.kt
* /opt/cola/permits/2100345168_1721125089.8386343/0/kotlin-sam-with-receiver-1-5-31-sources-jar/org/jetbrains/kotlin/noarg/gradle/SamWithReceiverSubplugin.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2016 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345168_1721125089.8386343/0/kotlin-sam-with-receiver-1-5-31-sources-jar/SamWithReceiverResolverExtension.kt
* /opt/cola/permits/2100345168_1721125089.8386343/0/kotlin-sam-with-receiver-1-5-31-sources-jar/SamWithReceiverPlugin.kt

# 1.222 rxjava 3.0.0

## 1.222.1 Available under license :

No license file was found, but licenses were detected in source scan.

/**
* Copyright (c) 2016-present, RxJava Contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in
* compliance with the License. You may obtain a copy of the License at

*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is
* distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
* the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSwitchIfEmpty.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeCount.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleOnErrorReturn.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableDoOnLifecycle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableJust.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableIgnoreElementsCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFlattenIterable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/util/NotificationLite.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableObserveOn.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableIntervalRange.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableTakeLast.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybePeek.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelPeek.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/ObservableOperator.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/FlowableOperator.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelConcatMap.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/fuseable/CancellableQueueFuseable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableGroupBy.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/AbstractMaybeWithUpstream.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableConcatMapEager.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/disposables/DisposableHelper.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/SimplePlainQueue.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeConcatArray.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/mixed/SingleFlatMapObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/subscribers/TestSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/parallel/ParallelFlowableConverter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/QueueFuseable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableRetryWhen.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableDelaySubscriptionOther.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableInterval.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableTakeUntilPredicate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/observers/QueueDrainObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/annotations/Nullable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFromAction.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSkipLast.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSkipLastTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeEmpty.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeMap.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableElementAt.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableRepeatUntil.java

* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeConcatIterable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFromCallable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableElementAtMaybePublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFromSupplier.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/Function5.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableDematerialize.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableFromCallable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/mixed/MaybeFlatMapObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableError.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/disposables/ActionDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleToFlowable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableMap.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/Notification.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/processors/SerializedProcessor.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableOnErrorReturn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/InnerQueuedObserverSupport.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableHide.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableDoFinally.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelFlatMapIterable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleDoOnLifecycle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/mixed/ObservableConcatMapSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/BackpressureStrategy.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/observers/BasicQueueDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableToObservable.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableElementAtMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableEmpty.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleTakeUntil.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableTimeoutTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleLift.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/HasUpstreamPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/observers/BlockingDisposableMultiObserver.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeToFlowable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableFromSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/SingleOperator.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableWithLatestFromMany.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableBuffer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleFlatMapCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeMergeArray.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/ObservableConverter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeDematerialize.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableCache.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFromSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/util/HalfSerializer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/subjects/UnicastSubject.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/schedulers/ImmediateThinScheduler.java

*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/subscribers/SafeSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/Observable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableObserveOn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/BiConsumerSingleObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleDelayWithCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscribers/InnerQueuedSubscriberSupport.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/MaybeConverter.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/fuseable/ScalarSupplier.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableDematerialize.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFromCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/subscribers/DisposableSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscriptions/EmptySubscription.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/processors/UnicastProcessor.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableScalarXMap.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableConcatWithCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableDefer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableConcatMapEagerPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableTake.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeIsEmptySingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFromIterable.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableZip.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/BooleanSupplier.java

* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableAll.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/mixed/ObservableSwitchMapSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableTimeout.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleToObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservablePublishSelector.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeToPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/schedulers/RxThreadFactory.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSampleTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleDoOnSuccess.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableAmb.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFlatMapNotification.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableReduceWithSingle.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFlatMapIterableFlowable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/BlockingFlowableMostRecent.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSerialized.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableTakeLastTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/schedulers/DisposeOnCancel.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableMergeDelayErrorIterable.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableCountSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/CompletableEmitter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleFromUnsafeSource.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableMergeWithCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/jdk8/ObservableFromStream.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleTimeout.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleMap.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/disposables/SubscriptionDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleDoOnTerminate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableThrottleFirstTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/jdk8/ObservableFirstStageObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/disposables/RunnableDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableJoin.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableCombineLatest.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/parallel/ParallelFailureHandling.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableElementAtSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/jdk8/MaybeMapOptional.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/HasUpstreamSingleSource.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFromSupplier.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableMapNotification.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSkipLast.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableCountSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableWithLatestFrom.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/subscribers/StrictSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableBufferExactBoundary.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleDematerialize.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSamplePublisher.java

*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/AbstractFlowableWithUpstream.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeDelayOtherPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableConcatMap.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeError.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/schedulers/NonBlockingThread.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelSortedJoin.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/subjects/Subject.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableFromUnsafeSource.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/BlockingFlowableNext.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSkip.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleMaterialize.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableConcatMapScheduler.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/util/QueueDrainHelper.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSingleSingle.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSequenceEqual.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeToSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelMap.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/observers/ResourceCompletableObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/CompletableFromCompletionStage.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableDistinctUntilChanged.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableWithLatestFromMany.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFlatMapCompletable.java

* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/flowables/ConnectableFlowable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableTakeLast.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/ObservableSource.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableWindowTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelJoin.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableSubscribeOn.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/mixed/MaybeFlatMapPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableRetryPredicate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleFlatMapMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/schedulers/Timed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableDelay.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableToListSingle.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableBlockingSubscribe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeLift.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableLastMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableReduceSeedSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/processors/FlowableProcessor.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/LambdaObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleHide.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/FlowableOnSubscribe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/functions/ObjectHelper.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscribers/BlockingBaseSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/observers/ResourceSingleObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelFilterTry.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableLift.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableMergeWithSingle.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableScan.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSkip.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/FlowableSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableTakeLastOne.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelRunOn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleJust.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleTimeInterval.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/mixed/CompletableAndThenPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/subscriptions/ArrayCompositeSubscription.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/processors/ReplayProcessor.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/mixed/FlowableConcatMapSinglePublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableWindowBoundarySelector.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/AbstractEmptyQueueFuseable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeDelayWithCompletable.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableCreate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleDoOnSubscribe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableInternalHelper.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableZipIterable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableScanSeed.java

* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeDoOnTerminate.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/queue/SpscLinkedArrayQueue.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/Emitter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableTakeUntilPredicate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFlatMapMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/Function8.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleFromCallable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/ParallelCollector.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableToList.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableTimer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableMergeArrayDelayError.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleDetach.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFlatten.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/FlowableFromCompletionStage.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/ObservableSingleStageObserver.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/FlowableLastStageSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableConcatArray.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableOnBackpressureBufferStrategy.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSubscribeOn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleFlatMapIterableFlowable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/BiFunction.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFromRunnable.java
*

/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableElementAtMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/BlockingLastObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableOnErrorNext.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFromFuture.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/Function.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscriptions/DeferredScalarSubscription.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/mixed/ObservableSwitchMapMaybe.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableAmb.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/schedulers/TestScheduler.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/observers/TestObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/MaybeOnSubscribe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFlatMapMaybePublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/util/AppendOnlyLinkedArrayList.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleZipArray.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableRefCount.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableToList.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/disposables/EmptyDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeIgnoreElement.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableOnErrorReturn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeMaterialize.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeDoFinally.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableDoAfterNext.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/operators/completable/CompletableConcat.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableNever.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableTimeInterval.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableOnErrorReturn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableDoOnEach.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/ConditionalSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/util/LinkedArrayList.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeConcatArrayDelayError.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/observers/CallbackCompletableObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFlatMapCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelFlatMap.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableWindowBoundarySelector.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableCollect.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeJust.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeNever.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableRetryPredicate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/disposables/FutureDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelReduce.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableReduceSeedSingle.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableDetach.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFilter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/util/BackpressureHelper.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/subjects/SerializedSubject.java

* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSingleMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableFromAction.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableDoAfterNext.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSkipWhile.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFilter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/BlockingFlowableLatest.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/subjects/ReplaySubject.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/Function9.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableWindow.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableElementAt.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/disposables/SequentialDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/Predicate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/util/QueueDrain.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscribers/BlockingSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableRange.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservablePublish.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeZipIterable.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/plugins/RxJavaPlugins.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableZip.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/Single.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/FlowableFlatMapStream.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeCallbackObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/operators/single/SingleCache.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableInternalHelper.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableOnBackpressureDrop.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/disposables/AutoCloseableDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFlatMapSinglePublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableHide.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableScan.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleDoOnDispose.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/jdk8/ParallelMapOptional.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/disposables/ReferenceDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/schedulers/ScheduledRunnable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/observers/BlockingMultiObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableUnsubscribeOn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/BlockingObservableLatest.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/FuseToFlowable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/subscribers/BasicFuseableConditionalSubscriber.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/mixed/FlowableSwitchMapSinglePublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/jdk8/FlowableFirstStageSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableReduceMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelFromArray.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleFlatMapNotification.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/HasUpstreamCompletableSource.java
*

/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableRepeat.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscribers/BlockingLastSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableDelaySubscriptionOther.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/ObservableLastStageObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/FlowableFromStream.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/disposables/CompositeDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/DeferredScalarObserver.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/FlowableSingleStageSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableReplay.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableRepeatWhen.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/BiPredicate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableBufferBoundary.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableDisposeOn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/IntFunction.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/SingleSource.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/Consumer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/schedulers/SingleScheduler.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/ResumeSingleObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/Maybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableOnErrorComplete.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/util/BlockingIgnoringReceiver.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeAmb.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/operators/observable/ObservableThrottleLatest.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/Flowable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleFromSupplier.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/subjects/AsyncSubject.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObserverResourceWrapper.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFromRunnable.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableDistinct.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFlattenIterable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/util/ExceptionHelper.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/util/SorterFunction.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableMergeWithCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableCollect.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableDebounceTimed.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/observers/BasicFuseableObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSkipUntil.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableRefCount.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableUsing.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableWindowSubscribeIntercept.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableBufferBoundary.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeTakeUntilPublisher.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/mixed/FlowableSwitchMapSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/subscribers/SubscriberResourceWrapper.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeSubscribeOn.java

* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeIgnoreElementCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/disposables/SerialDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/MaybeFlattenStreamAsFlowable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableAllSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeTimeInterval.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFlatMapSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFromAction.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelCollect.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSequenceEqualSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableRange.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFlatMapIterableObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeDelaySubscriptionOtherPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableConcatArray.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSingleMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFromFuture.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/SingleConverter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletablePeek.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFromCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/mixed/ObservableConcatMapCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/SingleFlattenStreamAsFlowable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFromArray.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSwitchIfEmpty.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/operators/observable/BlockingObservableIterable.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableGroupBy.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFlatMap.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowablePublish.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/CompletableOnSubscribe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/QueueSubscription.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/subscribers/BlockingFirstSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/Scheduler.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/SimpleQueue.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/annotations/Experimental.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/disposables/ListCompositeDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeDoAfterSuccess.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleCreate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/FuseToMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableCount.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeDoOnLifecycle.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/processors/BehaviorProcessor.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleDoOnError.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/annotations/CheckReturnValue.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableCreate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableTakeUntil.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableMergeWithMaybe.java
*

/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableFromRunnable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeOnErrorNext.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/mixed/ScalarXMapZHelper.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/mixed/FlowableSwitchMapCompletablePublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFlatMapSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/mixed/FlowableSwitchMapCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/CompletableConverter.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableOnErrorComplete.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/processors/PublishProcessor.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/MaybeOperator.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableBufferExactBoundary.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/FlowableEmitter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableTimeout.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableConcatMapScheduler.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleFromPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/disposables/CancellableDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/CompletableTransformer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableDoOnEvent.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableLastMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableMergeWithSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableTakePublisher.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableTakeLastTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableIgnoreElements.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelReduceFull.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/jdk8/MaybeFromCompletionStage.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableAutoConnect.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeEqualSingle.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/observers/BasicIntQueueDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeIsEmpty.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleEquals.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/QueueDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableWindowTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleFlatMapIterableObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/mixed/FlowableConcatMapMaybe.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleResumeNext.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSubscribeOn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableElementAtSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/util/EndConsumerHelper.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableUnsubscribeOn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/MaybeObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/functions/Supplier.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/jdk8/FlowableStageSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableDebounce.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/annotations/SchedulerSupport.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/jdk8/ParallelMapTryOptional.java

* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableDetach.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/ObservableFlatMapStream.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/ObservableStageObserver.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSerialized.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeDetach.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscribers/LambdaSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/schedulers/NewThreadWorker.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeSwitchIfEmptySingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/BlockingBaseObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableBufferTimed.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableReduce.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleNever.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableCreate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableMerge.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableFromObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/ObservableCollectWithCollectorSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/subjects/MaybeSubject.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableError.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/exceptions/OnErrorNotImplementedException.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/processors/AsyncProcessor.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/BlockingObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/SingleFlattenStreamAsObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/operators/completable/CompletableDelay.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableErrorSupplier.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/observers/DisposableCompletableObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFromFuture.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/subscribers/FutureSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFlatMapCompletableCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableWindowSubscribeIntercept.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/subjects/CompletableSubject.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeSwitchIfEmpty.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableObserveOn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableCount.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableLastSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/schedulers/SchedulerMultiWorkerSupport.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableOnErrorNext.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleDelayWithObservable.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/jdk8/SingleFromCompletionStage.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/disposables/Disposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/functions/Function3.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableBufferTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/ObservableEmitter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableLift.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/mixed/FlowableSwitchMapMaybePublisher.java
*

/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/queue/MpscLinkedQueue.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/ParallelFlatMapStream.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableInterval.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/ObservableTransformer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableZipIterable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableAndThenCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableDoFinally.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/observers/DisposableSingleObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeDefer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSwitchMap.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableThrottleLatest.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscriptions/SubscriptionHelper.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleDelay.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/Function6.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableRetryBiPredicate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/AbstractObservableWithUpstream.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/FlowableCollectWithCollectorSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableRepeatUntil.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFlatMapCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/observers/LambdaConsumerIntrospection.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableOnBackpressureBuffer.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/FutureMultiObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/util/VolatileSizeArrayList.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFlatMapSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/BlockingObservableMostRecent.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableToFlowable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeTimer.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/jdk8/FlowableCollectWithCollector.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleFlatMap.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/util/HashMapSupplier.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/util/OpenHashSet.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleSubscribeOn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleDelayWithSingle.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/subscribers/DefaultSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableAnySingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/util/ObservableQueueDrain.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableToSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/FuseToObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableMaterialize.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/util/MergerBiFunction.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableError.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableMaterialize.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableConcatMapEager.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFromAction.java

* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableLastSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeUsing.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/mixed/ObservableConcatMapMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableTakeUntil.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/parallel/ParallelFlowable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscribers/ForEachWhileSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableCombineLatest.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeOnErrorComplete.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/FlowableMapOptional.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableDistinct.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFromObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableIntervalRange.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/functions/Functions.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableUsing.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/Function4.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeZipArray.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/Completable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/EmptyCompletableObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleError.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableFromSupplier.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeTakeUntilMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/MaybeEmitter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/operators/observable/ObservableAnySingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/observables/ConnectableObservable.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableMaterialize.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleObserveOn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableAllSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/observables/GroupedObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeErrorCallable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableConcatWithMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableRangeLong.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSingleSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableRangeLong.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/schedulers/Schedulers.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleTimer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/mixed/FlowableConcatMapCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableEmpty.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/fuseable/HasUpstreamObservableSource.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/observers/ResourceObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableIgnoreElementsCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleFlatMapBiSelector.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableTake.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableWindowBoundary.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/CompletableObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/disposables/ArrayCompositeDisposable.java

*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/ObservableFromCompletionStage.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableMergeArray.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableConcatIterable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/ForEachWhileObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeCache.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFromIterable.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFlatMapMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/mixed/MaterializeSingleObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/SingleMapOptional.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeTimeoutMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/BiConsumer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableTakeUntilCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableMapNotification.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFromCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableAny.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableAmb.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleAmb.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/exceptions/UndeliverableException.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/Function7.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableReduceMaybe.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSkipWhile.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/SafeMaybeObserver.java

* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFromArray.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSkipLastTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleDefer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableCollectSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/util/ArrayListSupplier.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleDoAfterSuccess.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/processors/MulticastProcessor.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableTimer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/SafeSingleObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/CompletableSource.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableCache.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableMap.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableScanSeed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableConcatWithSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableIgnoreElements.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableConcatMap.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/MaybeSource.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscriptions/BasicIntQueueSubscription.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSequenceEqualSingle.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableTimeoutTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/subjects/BehaviorSubject.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/ObservableOnSubscribe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFromUnsafeSource.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/observers/DisposableMaybeObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/subscriptions/SubscriptionArbiter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/observers/SerializedObserver.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableDoOnLifecycle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleFlatMapPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/subscribers/BoundedSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableWindowBoundary.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSkipUntil.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableEmpty.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableHide.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeUnsubscribeOn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/exceptions/MissingBackpressureException.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/queue/SpscArrayQueue.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/observers/DisposableObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/util/EmptyComponent.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/subscribers/SinglePostCompleteSubscriber.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/annotations/BackpressureSupport.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFilterSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/SingleObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelMapTry.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableDefer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/subscribers/ResourceSubscriber.java

* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/MaybeTransformer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleUsing.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeObserveOn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableBuffer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeUnsafeCreate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/disposables/DisposableContainer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableDistinctUntilChanged.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/fuseable/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableResumeNext.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/mixed/ObservableSwitchMapCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableToListSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableConcatWithCompletable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/fuseable/HasUpstreamMaybeSource.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscribers/InnerQueuedSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/subjects/PublishSubject.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableDoOnEach.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFlatMapBiSelector.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableCollectSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/LongConsumer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/Observer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableOnBackpressureLatest.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelDoOnNextTry.java
*

/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeContains.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleOnErrorComplete.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableTakeWhile.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscriptions/ScalarSubscription.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableOnErrorComplete.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/CompletableOperator.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableDefer.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/flowables/GroupedFlowable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeOnErrorReturn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableConcatWithMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscriptions/AsyncSubscription.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/annotations/Beta.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableDebounceTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/mixed/FlowableSwitchMapMaybe.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFromRunnable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableWithLatestFrom.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleDoFinally.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableWindow.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscriptions/BooleanSubscription.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/util/BlockingHelper.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/subscribers/SerializedSubscriber.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/BlockingFlowableIterable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/operators/single/SingleUnsubscribeOn.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableConcatWithSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableAutoConnect.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableNever.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableRetryBiPredicate.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/BlockingObservableNext.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableDelay.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableMergeWithMaybe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/subscribers/DeferredScalarSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/SingleTransformer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableGenerate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/jdk8/ObservableMapOptional.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/observers/InnerQueuedObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableReplay.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleZipIterable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/mixed/CompletableAndThenObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFromPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableTimer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleInternalHelper.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/jdk8/MaybeFlattenStreamAsObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableCache.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableUsing.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFilter.java

* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSampleTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/Action.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/schedulers/ExecutorScheduler.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableDetach.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/util/SuppressAnimalSniffer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableTakeWhile.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/exceptions/ProtocolViolationException.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/observers/ResourceMaybeObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/util/ListAddBiConsumer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/ConsumerSingleObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableScalarXMap.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableLift.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSampleWithObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFromSupplier.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleDoAfterTerminate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/FlowableConverter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscriptions/BasicQueueSubscription.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/SingleEmitter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFromCallable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableTimeout.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeCreate.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableRetryWhen.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/subscribers/BasicFuseableSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/functions/Cancellable.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFlatMap.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeDelay.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableTakeLastOne.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/observers/FutureObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/completable/CompletableMergeIterable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableFromPublisher.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelFilter.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableMapPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/subjects/SingleSubject.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/annotations/NonNull.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/jdk8/CompletionStageConsumer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/util/ErrorMode.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/observers/DefaultObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/parallel/ParallelFromPublisher.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/observers/DeferredScalarDisposable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/observers/BaseTestConsumer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableAll.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/SingleOnSubscribe.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableSequenceEqual.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/single/SingleContains.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableRepeatWhen.java

*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleDelayWithPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableSwitchMap.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/jdk8/ObservableCollectWithCollector.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeToObservable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/exceptions/Exceptions.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/SafeCompletableObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableDoFinally.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/mixed/FlowableConcatMapMaybePublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/annotations/BackpressureKind.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableThrottleFirstTimed.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeTimeoutPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/util/AtomicThrowable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeDoOnEvent.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableNever.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/BlockingFirstObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/completable/CompletableFromPublisher.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/observers/SafeObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/subscribers/QueueDrainSubscriber.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/FlowableTransformer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableFlatMapCompletableCompletable.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableJust.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableDebounce.java

* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/single/SingleDoOnEvent.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/parallel/ParallelTransformer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeFromCallable.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/maybe/MaybeHide.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableBlockingSubscribe.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableReduceWithSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/mixed/FlowableConcatMapSingle.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableTimeInterval.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/observers/DisposableLambdaObserver.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/util/ConnectConsumer.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableRepeat.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableAny.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableGenerate.java
No license file was found, but licenses were detected in source scan.

/**
* Copyright (c) 2016-present, RxJava Contributors.
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in
* compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software distributed under the License is
* distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
* the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/schedulers/SchedulerRunnableIntrospection.java
No license file was found, but licenses were detected in source scan.

/**
* Copyright (c) 2016-present, RxJava Contributors.

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/annotations/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/exceptions/CompositeException.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/observables/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/schedulers/IoScheduler.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/processors/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/schedulers/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/schedulers/ComputationScheduler.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/core/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableGroupJoin.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/observable/ObservableJoin.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/exceptions/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/operators/flowable/FlowablePublishMulticast.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/functions/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/disposables/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/plugins/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-

jar/io/reactivex/rxjava3/internal/schedulers/SchedulerWhen.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/core/BackpressureOverflowStrategy.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/schedulers/NewThreadScheduler.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/observers/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/schedulers/SchedulerPoolFactory.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/subjects/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/subscribers/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/parallel/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/flowables/package-info.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/schedulers/TrampolineScheduler.java
*
/opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/observable/ObservableGroupJoin.java
No license file was found, but licenses were detected in source scan.

/**
* Copyright (c) 2016-present, RxJava Contributors.
* <p>
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in
* compliance with the License. You may obtain a copy of the License at
* <p>
* http://www.apache.org/licenses/LICENSE-2.0
* <p>
* Unless required by applicable law or agreed to in writing, software distributed under the License is
* distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
or implied. See
* the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-
jar/io/reactivex/rxjava3/internal/operators/flowable/FlowableOnBackpressureError.java
No license file was found, but licenses were detected in source scan.

/**
* Copyright (c) 2016-present, RxJava Contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/schedulers/ScheduledDirectTask.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/schedulers/InstantPeriodicTask.java
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/schedulers/AbstractDirectTask.java
*
 /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/schedulers/ScheduledDirectPeriodicTask.java
No license file was found, but licenses were detected in source scan.

/**
* Copyright (c) 2016-present, RxJava Contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in
* compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is
* distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
* the License for the specific language governing permissions and limitations under the License.
*/
/*
* Original License: https://github.com/JCTools/JCTools/blob/master/LICENSE
* Original location: https://github.com/JCTools/JCTools/blob/master/jctools-core/src/main/java/org/jctools/util/Pow2.java
*/

Found in path(s):
* /opt/cola/permits/2100344934_1721125169.0185564/0/rxjava-3-0-0-sources-jar/io/reactivex/rxjava3/internal/util/Pow2.java

# 1.223 hilt-android-gradle-plugin 2.44

## 1.223.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2022 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/util/Files.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/AndroidEntryPointClassVisitor.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/util/Configurations.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/util/Tasks.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/util/CopyTransform.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/root/AggregatedAnnotation.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/root/ComponentTreeDepsGenerator.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/root/ProcessedRootSentinelGenerator.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/root/AggregatedElementProxyGenerator.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/util/AggregatedPackagesTransform.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/task/AggregateDepsTask.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/root/Aggregator.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/HiltGradlePlugin.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/AndroidEntryPointClassTransformer.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/AndroidEntryPointTransform.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/task/HiltTransformTestClassesTask.kt

* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/HiltExtension.kt
* /opt/cola/permits/2100345334_1721124779.8508673/0/hilt-android-gradle-plugin-2-44-sources-jar/dagger/hilt/android/plugin/util/Strings.kt

# 1.224 kotlin-scripting-compiler-impl-embeddable 1.7.10

## 1.224.1 Available under license :

Apache-2.0

# 1.225 kotlin-script-runtime 1.5.31

## 1.225.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345141_1721124865.955726/0/kotlin-script-runtime-1-5-31-sources-2-jar/kotlin/script/experimental/dependencies/resolvers.kt
* /opt/cola/permits/2100345141_1721124865.955726/0/kotlin-script-runtime-1-5-31-sources-2-jar/kotlin/script/dependencies/dependencies_deprecated.kt
* /opt/cola/permits/2100345141_1721124865.955726/0/kotlin-script-runtime-1-5-31-sources-2-jar/kotlin/script/dependencies/resolvers_deprecated.kt
* /opt/cola/permits/2100345141_1721124865.955726/0/kotlin-script-runtime-1-5-31-sources-2-jar/kotlin/script/experimental/dependencies/dependencies.kt
* /opt/cola/permits/2100345141_1721124865.955726/0/kotlin-script-runtime-1-5-31-sources-2-jar/kotlin/script/extensions/samWithReceiver.kt
* /opt/cola/permits/2100345141_1721124865.955726/0/kotlin-script-runtime-1-5-31-sources-2-

jar/kotlin/script/experimental/dependencies/AsyncDependenciesResolver.kt
* /opt/cola/permits/2100345141_1721124865.955726/0/kotlin-script-runtime-1-5-31-sources-2-jar/kotlin/script/templates/annotations.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2016 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345141_1721124865.955726/0/kotlin-script-runtime-1-5-31-sources-2-jar/kotlin/script/templates/standard/templates.kt

# 1.226 kotlin-scripting-jvm 1.5.31

## 1.226.1 Available under license :

Apache-2.0

# 1.227 gradle-kotlin-dsl-plugins 2.1.7

## 1.227.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2021 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
```

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344864_1721145701.229218/0/gradle-kotlin-dsl-plugins-2-1-7-sources-jar/org/gradle/kotlin/dsl/plugins/dsl/ExperimentalCompilerWarningSilencer.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344864_1721145701.229218/0/gradle-kotlin-dsl-plugins-2-1-7-sources-jar/org/gradle/kotlin/dsl/plugins/embedded/EmbeddedKotlinPlugin.kt
* /opt/cola/permits/2100344864_1721145701.229218/0/gradle-kotlin-dsl-plugins-2-1-7-sources-jar/org/gradle/kotlin/dsl/plugins/dsl/KotlinDslPlugin.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344864_1721145701.229218/0/gradle-kotlin-dsl-plugins-2-1-7-sources-jar/org/gradle/kotlin/dsl/plugins/Version.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2018 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344864_1721145701.229218/0/gradle-kotlin-dsl-plugins-2-1-7-sources-jar/org/gradle/kotlin/dsl/plugins/dsl/KotlinDslCompilerPlugins.kt
* /opt/cola/permits/2100344864_1721145701.229218/0/gradle-kotlin-dsl-plugins-2-1-7-sources-jar/org/gradle/kotlin/dsl/plugins/dsl/KotlinDslPluginOptions.kt
* /opt/cola/permits/2100344864_1721145701.229218/0/gradle-kotlin-dsl-plugins-2-1-7-sources-jar/org/gradle/kotlin/dsl/plugins/precompiled/PrecompiledScriptPlugins.kt
* /opt/cola/permits/2100344864_1721145701.229218/0/gradle-kotlin-dsl-plugins-2-1-7-sources-jar/org/gradle/kotlin/dsl/plugins/base/KotlinDslBasePlugin.kt

# 1.228 kotlin-android-extensions 1.7.10

## 1.228.1 Available under license :

Apache-2.0

# 1.229 kotlin-reflect 1.8.0

## 1.229.1 Available under license :

Apache-2.0

# 1.230 jackson-module-kotlin 2.12.4

## 1.230.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

# 1.231 kotlin-stdlib 1.7.0

## 1.231.1 Available under license :

Apache-2.0

# 1.232 protos 30.3.0

## 1.232.1 Available under license :

Copyright (c) 2005-2016, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited
to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form
of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.233 tracker 30.3.0

## 1.233.1 Available under license :

Copyright (c) 2005-2016, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.234 annotations 30.3.0

## 1.234.1 Available under license :

Copyright (c) 2005-2013, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited
to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated
within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.235 sdklib 30.3.0

## 1.235.1 Available under license :

Copyright (c) 2005-2008, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.236 collection 1.0.0

## 1.236.1 Available under license :

MIT License

Copyright (c) 2018 Pol Dellaiera

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.237 kotlin-script-runtime 1.7.10

## 1.237.1 Available under license :

Apache-2.0

# 1.238 kotlin-util-io 1.5.31

## 1.238.1 Available under license :

Apache-2.0

# 1.239 kotlinx-coroutines-core-jvm 1.6.1

## 1.239.1 Available under license :

Apache-2.0

# 1.240 kotlin-gradle-plugin-idea 1.7.10

## 1.240.1 Available under license :

Apache-2.0

# 1.241 zipflinger 7.3.0

## 1.241.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2021 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
```

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/Sources.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/CentralDirectoryRecord.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/ZipSourceEntry.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/CentralDirectory.java
*
 /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/FreeStore.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/Ints.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/Location.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/Source.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/Entry.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-

jar/com/android/zipflinger/LocalFileHeader.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/Compressor.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/ZipMap.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/ExtractionInfo.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/ZipReader.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/EndOfCentralDirectory.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/ZipInfo.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/ZipWriter.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/Crc32.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/ZipSource.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/BytesSource.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/NoCopyByteArrayOutputStream.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/Archive.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/ZipArchive.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/StableArchive.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/SynchronizedArchive.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/Zip64Locator.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/Zip64Eocd.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/ZipSourceEntryPipe.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/FullFileSource.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/PayloadInputStream.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/Zip64.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/LargeFileSource.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/ZipRepo.java
* /opt/cola/permits/2100345153_1721144709.3735135/0/zipflinger-7-3-0-sources-jar/com/android/zipflinger/StreamSource.java

# 1.242 kotlin-gradle-plugin-api 1.5.31

## 1.242.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2014 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345045_1721124834.6802068/0/kotlin-gradle-plugin-api-1-5-31-sources-

jar/org/jetbrains/kotlin/gradle/plugin/KotlinGradleSubplugin.kt

# 1.243 gradle-test-logger-plugin 1.6.0

## 1.243.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{}"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.244 kotlin-scripting-compiler-embeddable 1.7.10

## 1.244.1 Available under license :

Apache-2.0

# 1.245 hilt-core 2.44

## 1.245.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/components/package-info.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/internal/GeneratedComponentManagerHolder.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/codegen/package-info.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/package-info.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/migration/DisableInstallInCheck.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/components/SingletonComponent.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/internal/Preconditions.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/migration/AliasOf.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/codegen/OriginatingElement.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/internal/aliasof/AliasOfPropagatedData.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/migration/package-info.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/internal/UnsafeCasts.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/internal/definecomponent/DefineComponentClasses.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/EntryPoint.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/DefineComponent.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/processor/internal/aggregateddeps/AggregatedDeps.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/internal/GeneratedComponentManager.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/internal/generatesrootinput/GeneratesRootInputPropagatedData.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/GeneratesRootInput.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/internal/GeneratedComponent.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/EntryPoints.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/InstallIn.java
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/internal/definecomponent/DefineComponentNoParent.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344662_1721124985.7724292/0/hilt-core-2-44-sources-jar/dagger/hilt/internal/TestSingletonComponent.java

# 1.246 fastutil 8.4.0

## 1.246.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2009-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*
*
*
* Copyright (C) 1999 CERN - European Organization for Nuclear Research.
*
*   Permission to use, copy, modify, distribute and sell this software and
*   its documentation for any purpose is hereby granted without fee,
*   provided that the above copyright notice appear in all copies and that
*   both that copyright
 notice and this permission notice appear in
*   supporting documentation. CERN makes no representations about the
*   suitability of this software for any purpose. It is provided "as is"
*   without expressed or implied warranty.
*/

package PACKAGE;

import java.util.Arrays;
import java.util.Random;

import it.unimi.dsi.fastutil.BigArrays;
import it.unimi.dsi.fastutil.Hash;
import static it.unimi.dsi.fastutil.BigArrays.ensureLength;
import static it.unimi.dsi.fastutil.BigArrays.start;
import static it.unimi.dsi.fastutil.BigArrays.segment;
import static it.unimi.dsi.fastutil.BigArrays.displacement;
import static it.unimi.dsi.fastutil.BigArrays.SEGMENT_MASK;
import static it.unimi.dsi.fastutil.BigArrays.SEGMENT_SHIFT;
```

```
import static it.unimi.dsi.fastutil.BigArrays.SEGMENT_SIZE;

#if KEYS_PRIMITIVE
#if KEY_CLASS_Integer
import java.util.concurrent.atomic.AtomicIntegerArray;
#endif
#if KEY_CLASS_Long
import java.util.concurrent.atomic.AtomicLongArray;
#endif

#if ! KEY_CLASS_Byte
 && ! KEY_CLASS_Boolean
import it.unimi.dsi.fastutil.bytes.ByteBigArrays;
#endif

/** A class providing static methods and objects that do useful things with {@linkplain BigArrays big arrays}.
*
* <p><strong>WARNING</strong>: as of 8.3.0, many methods such as {@link #length()} have been moved to
* {@link BigArrays}, as they can be imported statically in a more convenient way.
*
* <p>Note that {@link it.unimi.dsi.fastutil.io.BinIO} and {@link it.unimi.dsi.fastutil.io.TextIO}
* contain several methods that make it possible to load and save big arrays of primitive types as sequences
* of elements in {@link java.io.DataInput} format (i.e., not as objects) or as sequences of lines of text.
*
* @see BigArrays
*/

public final class BIG_ARRAYS {

#else

import java.util.Comparator;

/** A class providing static methods and objects that do useful things with {@linkplain BigArrays big arrays}.
*
* <p>Since version 8.4.0, many methods previously in this class(e.g., <code>length()</code>)
 have
* been deprecated in favor of those in {@link BigArrays}, which can be imported statically.
*
* <p>Note that {@link it.unimi.dsi.fastutil.io.BinIO} and {@link it.unimi.dsi.fastutil.io.TextIO}
* contain several methods make it possible to load and save big arrays of primitive types as sequences
* of elements in {@link java.io.DataInput} format (i.e., not as objects) or as sequences of lines of text.
*
* <p><strong>Warning:</strong> creating arrays
* using {@linkplain java.lang.reflect.Array#newInstance(Class,int) reflection}, as it
* happens in {@link #ensureCapacity(Object[][],long,long)} and {@link #grow(Object[][],long,long)},
* is <em>significantly slower</em> than using {@code new}. This phenomenon is particularly
* evident in the first growth phases of an array reallocated with doubling (or similar) logic.
```

```
*
* @see BigArrays
*/

public final class BIG_ARRAYS {

#endif
 private BIG_ARRAYS() {}

 /** A static, final, empty big array. */
 public static final
 KEY_TYPE[][] EMPTY_BIG_ARRAY = {};

 /** A static, final, empty big array to be used as default big array in allocations. An
   * object distinct from {@link #EMPTY_BIG_ARRAY} makes it possible to have different
   * behaviors depending on whether the user required an empty allocation, or we are
   * just lazily delaying allocation.
   *
   * @see java.util.ArrayList
   */
 public static final KEY_TYPE[][] DEFAULT_EMPTY_BIG_ARRAY = {};

 /** Returns the element of the given big array of specified index.
  *
  * @param array a big array.
  * @param index a position in the big array.
  * @return the element of the big array at the specified position.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
  */
 @Deprecated
 public static KEY_GENERIC KEY_GENERIC_TYPE get(final KEY_GENERIC_TYPE[][] array, final long index)
{
  return array[segment(index)][displacement(index)];
 }

 /** Sets the element of the given big array of specified index.
  *
  * @param array
 a big array.
  * @param index a position in the big array.
  * @param value the new value for the array element at the specified position.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
  */
 @Deprecated
 public static KEY_GENERIC void set(final KEY_GENERIC_TYPE[][] array, final long index,
KEY_GENERIC_TYPE value) {
  array[segment(index)][displacement(index)] = value;
 }
```

```
/** Swaps the element of the given big array of specified indices.
 *
 * @param array a big array.
 * @param first a position in the big array.
 * @param second a position in the big array.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
 */
@Deprecated
public static KEY_GENERIC void swap(final KEY_GENERIC_TYPE[][] array, final long first, final long second)
{
 final KEY_GENERIC_TYPE t = array[segment(first)][displacement(first)];
 array[segment(first)][displacement(first)] = array[segment(second)][displacement(second)];
 array[segment(second)][displacement(second)]
= t;
}

#if KEYS_PRIMITIVE && ! KEY_CLASS_Boolean
/** Adds the specified increment the element of the given big array of specified index.
 *
 * @param array a big array.
 * @param index a position in the big array.
 * @param incr the increment
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
 */
@Deprecated
public static void add(final KEY_GENERIC_TYPE[][] array, final long index, KEY_GENERIC_TYPE incr) {
 array[segment(index)][displacement(index)] += incr;
}

/** Multiplies by the specified factor the element of the given big array of specified index.
 *
 * @param array a big array.
 * @param index a position in the big array.
 * @param factor the factor
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
 */
@Deprecated
public static void mul(final KEY_GENERIC_TYPE[][] array, final long index, KEY_GENERIC_TYPE factor) {
 array[segment(index)][displacement(index)] *=
factor;
}

/** Increments the element of the given big array of specified index.
 *
 * @param array a big array.
 * @param index a position in the big array.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
```

```
 */
@Deprecated
public static void incr(final KEY_GENERIC_TYPE[][] array, final long index) {
 array[segment(index)][displacement(index)]++;
}

/** Decrements the element of the given big array of specified index.
 *
 * @param array a big array.
 * @param index a position in the big array.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
 */
@Deprecated
public static void decr(final KEY_GENERIC_TYPE[][] array, final long index) {
 array[segment(index)][displacement(index)]--;
}


#endif


/** Returns the length of the given big array.
 *
 * @param array a big array.
 * @return the length of the given big array.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.

 */
@Deprecated
public static KEY_GENERIC long length(final KEY_GENERIC_TYPE[][] array) {
 final int length = array.length;
 return length == 0 ? 0 : start(length - 1) + array[length - 1].length;
}

/** Copies a big array from the specified source big array, beginning at the specified position, to the specified
position of the destination big array.
 * Handles correctly overlapping regions of the same big array.
 *
 * @param srcArray the source big array.
 * @param srcPos the starting position in the source big array.
 * @param destArray the destination big array.
 * @param destPos the starting position in the destination data.
 * @param length the number of elements to be copied.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
 */
@Deprecated
public static KEY_GENERIC void copy(final KEY_GENERIC_TYPE[][] srcArray, final long srcPos, final
KEY_GENERIC_TYPE[][] destArray, final long destPos, long length) {
```

```
 BigArrays.copy(srcArray, srcPos,
 destArray, destPos, length);
 }

 /** Copies a big array from the specified source big array, beginning at the specified position, to the specified
position of the destination array.
  *
  * @param srcArray the source big array.
  * @param srcPos the starting position in the source big array.
  * @param destArray the destination array.
  * @param destPos the starting position in the destination data.
  * @param length the number of elements to be copied.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
  */
 @Deprecated
 public static KEY_GENERIC void copyFromBig(final KEY_GENERIC_TYPE[][] srcArray, final long srcPos, final
KEY_GENERIC_TYPE[] destArray, int destPos, int length) {
  BigArrays.copyFromBig(srcArray, srcPos, destArray, destPos, length);
 }

 /** Copies an array from the specified source array, beginning at the specified position, to the specified position of
the destination big array.
  *
  * @param srcArray the source array.
  * @param
 srcPos the starting position in the source array.
  * @param destArray the destination big array.
  * @param destPos the starting position in the destination data.
  * @param length the number of elements to be copied.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
  */
 @Deprecated
 public static KEY_GENERIC void copyToBig(final KEY_GENERIC_TYPE[] srcArray, int srcPos, final
KEY_GENERIC_TYPE[][] destArray, final long destPos, long length) {
  BigArrays.copyToBig(srcArray, srcPos, destArray, destPos, length);
 }

#if KEY_CLASS_Object
 /** Creates a new big array using the given one as prototype.
  *
  * <p>This method returns a new big array of the given length whose element
  * are of the same class as of those of {@code prototype}. In case
  * of an empty big array, it tries to return {@link #EMPTY_BIG_ARRAY}, if possible.
  *
  * @param prototype a big array that will be used to type the new one.
  * @param length the length of the new big array.
```

```
 * @return a new big array of given type and length.
  */

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public static <K> K[][] newBigArray(final K[][] prototype, final long length) {
  return (K[][])newBigArray(prototype.getClass().getComponentType(), length);
 }

 /** Creates a new big array using a given component type.
  *
  * <p>This method returns a new big array whose segments
  * are of class {@code componentType}. In case
  * of an empty big array, it tries to return {@link #EMPTY_BIG_ARRAY}, if possible.
  *
  * @param componentType a class representing the type of segments of the array to be created.
  * @param length the length of the new big array.
  * @return a new big array of given type and length.
  */

 public static Object[][] newBigArray(Class<?> componentType, final long length) {
  if (length == 0 && componentType == Object[].class) return EMPTY_BIG_ARRAY;
  ensureLength(length);
  final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
  Object[][] base
 = (Object[][])java.lang.reflect.Array.newInstance(componentType, baseLength);
  final int residual = (int)(length & SEGMENT_MASK);
  if (residual != 0) {
   for(int i = 0; i < baseLength - 1; i++) base[i] =
(Object[])java.lang.reflect.Array.newInstance(componentType.getComponentType(), SEGMENT_SIZE);
   base[baseLength - 1] = (Object[])java.lang.reflect.Array.newInstance(componentType.getComponentType(),
residual);
  }
  else for(int i = 0; i < baseLength; i++) base[i] =
(Object[])java.lang.reflect.Array.newInstance(componentType.getComponentType(), SEGMENT_SIZE);

  return base;
 }
#endif

 /** Creates a new big array.
  *
  * @param length the length of the new big array.
  * @return a new big array of given length.
  */

 public static KEY_TYPE[][] newBigArray(final long length) {
  if (length == 0) return EMPTY_BIG_ARRAY;
  ensureLength(length);
```

```
 final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
 KEY_TYPE[][] base = new KEY_TYPE[baseLength][];
 final int
residual = (int)(length & SEGMENT_MASK);
 if (residual != 0) {
  for(int i = 0; i < baseLength - 1; i++) base[i] = new KEY_TYPE[SEGMENT_SIZE];
  base[baseLength - 1] = new KEY_TYPE[residual];
 }
 else for(int i = 0; i < baseLength; i++) base[i] = new KEY_TYPE[SEGMENT_SIZE];

 return base;
}

#if KEY_CLASS_Long || KEY_CLASS_Integer
 /** A static, final, empty big atomic array. */
 public static final ATOMIC_ARRAY[] EMPTY_BIG_ATOMIC_ARRAY = {};

 /** Creates a new big atomic array.
  *
  * @param length the length of the new big array.
  * @return a new big atomic array of given length.
  */

 public static ATOMIC_ARRAY[] newBigAtomicArray(final long length) {
  if (length == 0) return EMPTY_BIG_ATOMIC_ARRAY;
  ensureLength(length);
  final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
  ATOMIC_ARRAY[] base = new ATOMIC_ARRAY[baseLength];
  final int residual = (int)(length & SEGMENT_MASK);
  if (residual != 0) {
   for(int i = 0; i < baseLength - 1; i++) base[i]
 = new ATOMIC_ARRAY(SEGMENT_SIZE);
   base[baseLength - 1] = new ATOMIC_ARRAY(residual);
  }
  else for(int i = 0; i < baseLength; i++) base[i] = new ATOMIC_ARRAY(SEGMENT_SIZE);

  return base;
 }
#endif

#if KEY_CLASS_Object
 /** Turns a standard array into a big array.
  *
  * <p>Note that the returned big array might contain as a segment the original array.
  *
  * @param array an array.
  * @return a new big array with the same length and content of {@code array}.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
```

```
 */
 @Deprecated
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public static <K> K[][] wrap(final K[] array) {
  return BigArrays.wrap(array);
 }

#else
 /** Turns a standard array into a big array.
  *
  * <p>Note that the returned big array might contain as a segment the original array.
  *
  * @param array an array.
  * @return a new big array with the same length and content of {@code array}.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.

 */
 @Deprecated
 public static KEY_TYPE[][] wrap(final KEY_TYPE[] array) {
  return BigArrays.wrap(array);
 }
#endif
 /** Ensures that a big array can contain the given number of entries.
  *
  * <p>If you cannot foresee whether this big array will need again to be
  * enlarged, you should probably use {@code grow()} instead.
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new minimum length for this big array.
  * @return {@code array}, if it contains {@code length} entries or more; otherwise,
  * a big array with {@code length} entries whose first {@code length(array)}
  * entries are the same as those of {@code array}.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
  */
 @Deprecated
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] ensureCapacity(final
 KEY_GENERIC_TYPE[][] array, final long length) {
  return ensureCapacity(array, length, length(array));
 }

#if KEY_CLASS_Object

 /** Forces a big array to contain the given number of entries, preserving just a part of the big array.
  *
  * <p>This method returns a new big array of the given length whose element
```

```
 * are of the same class as of those of {@code array}.
 *
 * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
 * array, which must be considered read-only after calling this method.
 *
 * @param array a big array.
 * @param length the new minimum length for this big array.
 * @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.
 * @return a big array with {@code length} entries whose first {@code preserve}
 * entries are the same as those of {@code array}.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.

 */
 @Deprecated
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] forceCapacity(final KEY_GENERIC_TYPE[][] array,
final long length, final long preserve) {
  return BigArrays.forceCapacity(array, length, preserve);
 }

 /** Ensures that a big array can contain the given number of entries, preserving just a part of the big array.
 *
 * <p>This method returns a new big array of the given length whose element
 * are of the same class as of those of {@code array}.
 *
 * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
 * array, which must be considered read-only after calling this method.
 *
 * @param array a big array.
 * @param length the new minimum length for this big array.
 * @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.
 * @return {@code array}, if it can contain {@code length} entries or more; otherwise,
 * a big array with
 {@code length} entries whose first {@code preserve}
 * entries are the same as those of {@code array}.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
 */
 @Deprecated
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] ensureCapacity(final KEY_GENERIC_TYPE[][] array,
final long length, final long preserve) {
  return length > length(array) ? forceCapacity(array, length, preserve) : array;
 }

#else
```

```
 /** Forces a big array to contain the given number of entries, preserving just a part of the big array.
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new minimum length for this big array.
  * @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.
  * @return a big array with
 {@code length} entries whose first {@code preserve}
  * entries are the same as those of {@code array}.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
  */
 @Deprecated
 public static KEY_TYPE[][] forceCapacity(final KEY_TYPE[][] array, final long length, final long preserve) {
  return BigArrays.forceCapacity(array, length, preserve);
 }

 /** Ensures that a big array can contain the given number of entries, preserving just a part of the big array.
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new minimum length for this big array.
  * @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.
  * @return {@code array}, if it can contain {@code length} entries or more; otherwise,
  * a big array with {@code length}
 entries whose first {@code preserve}
  * entries are the same as those of {@code array}.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
  */
 @Deprecated
 public static KEY_TYPE[][] ensureCapacity(final KEY_TYPE[][] array, final long length, final long preserve) {
  return length > length(array) ? forceCapacity(array, length, preserve) : array;
 }

#endif

 /** Grows the given big array to the maximum between the given length and
  * the current length increased by 50%, provided that the given
  * length is larger than the current length.
  *
  * <p>If you want complete control on the big array growth, you
  * should probably use {@code ensureCapacity()} instead.
  *
```

```
 * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
 * array, which must be considered read-only after calling this method.
 *
 * @param array a big array.
 * @param length the new minimum length for this big array.
 * @return {@code
array}, if it can contain {@code length}
 * entries; otherwise, a big array with
 * max({@code length},{@code length(array)}/&phi;) entries whose first
 * {@code length(array)} entries are the same as those of {@code array}.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
 */
@Deprecated
public static KEY_GENERIC KEY_GENERIC_TYPE[][] grow(final KEY_GENERIC_TYPE[][] array, final long
length) {
 final long oldLength = length(array);
 return length > oldLength ? grow(array, length, oldLength) : array;
}

/** Grows the given big array to the maximum between the given length and
 * the current length increased by 50%, provided that the given
 * length is larger than the current length, preserving just a part of the big array.
 *
 * <p>If you want complete control on the big array growth, you
 * should probably use {@code ensureCapacity()} instead.
 *
 * <p><strong>Warning:</strong> the returned array might use part of the segments of
the original
 * array, which must be considered read-only after calling this method.
 *
 * @param array a big array.
 * @param length the new minimum length for this big array.
 * @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.
 * @return {@code array}, if it can contain {@code length}
 * entries; otherwise, a big array with
 * max({@code length},{@code length(array)}/&phi;) entries whose first
 * {@code preserve} entries are the same as those of {@code array}.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
 */
@Deprecated
public static KEY_GENERIC KEY_GENERIC_TYPE[][] grow(final KEY_GENERIC_TYPE[][] array, final long
length, final long preserve) {
 final long oldLength = length(array);
 return length > oldLength ? ensureCapacity(array, Math.max(oldLength + (oldLength >> 1), length), preserve) :
array;
}
```

```
#if KEY_CLASS_Object

 /** Trims the given big array to the given
 length.
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new maximum length for the big array.
  * @return {@code array}, if it contains {@code length}
  * entries or less; otherwise, a big array with
  * {@code length} entries whose entries are the same as
  * the first {@code length} entries of {@code array}.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
  */
 @Deprecated
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] trim(final KEY_GENERIC_TYPE[][] array, final long
length) {
  return BigArrays.trim(array, length);
 }

#else

 /** Trims the given big array to the given length.
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param
 array a big array.
  * @param length the new maximum length for the big array.
  * @return {@code array}, if it contains {@code length}
  * entries or less; otherwise, a big array with
  * {@code length} entries whose entries are the same as
  * the first {@code length} entries of {@code array}.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
  */
 @Deprecated
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] trim(final KEY_GENERIC_TYPE[][] array, final long
length) {
  ensureLength(length);
  final long oldLength = length(array);
  if (length >= oldLength) return array;
  final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
  final KEY_TYPE[][] base = Arrays.copyOf(array, baseLength);
  final int residual = (int)(length & SEGMENT_MASK);
  if (residual != 0) base[baseLength - 1] = ARRAYS.trim(base[baseLength - 1], residual);
```

```
 return base;
 }

#endif

 /** Sets the length of the given big array.
  *
  * <p><strong>Warning:</strong>
 the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new length for the big array.
  * @return {@code array}, if it contains exactly {@code length}
  * entries; otherwise, if it contains <em>more</em> than
  * {@code length} entries, a big array with {@code length} entries
  * whose entries are the same as the first {@code length} entries of
  * {@code array}; otherwise, a big array with {@code length} entries
  * whose first {@code length(array)} entries are the same as those of
  * {@code array}.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
  */
 @Deprecated
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] setLength(final KEY_GENERIC_TYPE[][] array, final
long length) {
  return BigArrays.setLength(array, length);
 }

 /** Returns a copy of a portion of a big array.
  *
  * @param array a big array.
  * @param
 offset the first element to copy.
  * @param length the number of elements to copy.
  * @return a new big array containing {@code length} elements of {@code array} starting at {@code offset}.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
  */
 @Deprecated
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] copy(final KEY_GENERIC_TYPE[][] array, final long
offset, final long length) {
  return BigArrays.copy(array, offset, length);
 }

 /** Returns a copy of a big array.
  *
  * @param array a big array.
  * @return a copy of {@code array}.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
```

```
 */
@Deprecated
public static KEY_GENERIC KEY_GENERIC_TYPE[][] copy(final KEY_GENERIC_TYPE[][] array) {
 return BigArrays.copy(array);
}

/** Fills the given big array with the given value.
 *
 * <p>This method uses a backward loop. It is significantly faster than the corresponding
 * method in {@link java.util.Arrays}.

*
 * @param array a big array.
 * @param value the new value for all elements of the big array.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
 */
@Deprecated
public static KEY_GENERIC void fill(final KEY_GENERIC_TYPE[][] array, final KEY_GENERIC_TYPE value)
{
 for(int i = array.length; i-- != 0;) Arrays.fill(array[i], value);
}

/** Fills a portion of the given big array with the given value.
 *
 * <p>If possible (i.e., {@code from} is 0) this method uses a
 * backward loop. In this case, it is significantly faster than the
 * corresponding method in {@link java.util.Arrays}.
 *
 * @param array a big array.
 * @param from the starting index of the portion to fill.
 * @param to the end index of the portion to fill.
 * @param value the new value for all elements of the specified portion of the big array.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
 */
@Deprecated
public static KEY_GENERIC void
fill(final KEY_GENERIC_TYPE[][] array, final long from, long to, final KEY_GENERIC_TYPE value) {
 BigArrays.fill(array, from, to, value);
}


/** Returns true if the two big arrays are elementwise equal.
 *
 * <p>This method uses a backward loop. It is significantly faster than the corresponding
 * method in {@link java.util.Arrays}.
 *
 * @param a1 a big array.
 * @param a2 another big array.
```

```
 * @return true if the two big arrays are of the same length, and their elements are equal.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
 */
@Deprecated
public static KEY_GENERIC boolean equals(final KEY_GENERIC_TYPE[][] a1, final KEY_GENERIC_TYPE
a2[][]) {
 return BigArrays.equals(a1, a2);
}

/* Returns a string representation of the contents of the specified big array.
 *
 * The string representation consists of a list of the big array's elements, enclosed in square brackets ("[]"). Adjacent
elements are separated by the characters ", " (a
comma followed by a space). Returns "null" if {@code a} is null.
 * @param a the big array whose string representation to return.
 * @return the string representation of {@code a}.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
 */
@Deprecated
public static KEY_GENERIC String toString(final KEY_GENERIC_TYPE[][] a) {
 return BigArrays.toString(a);
}


/** Ensures that a range given by its first (inclusive) and last (exclusive) elements fits a big array.
 *
 * <p>This method may be used whenever a big array range check is needed.
 *
 * @param a a big array.
 * @param from a start index (inclusive).
 * @param to an end index (inclusive).
 * @throws IllegalArgumentException if {@code from} is greater than {@code to}.
 * @throws ArrayIndexOutOfBoundsException if {@code from} or {@code to} are greater than the big array
length or negative.
 * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
 */
@Deprecated
public
static KEY_GENERIC void ensureFromTo(final KEY_GENERIC_TYPE[][] a, final long from, final long to) {
 BigArrays.ensureFromTo(length(a), from, to);
}

/** Ensures that a range given by an offset and a length fits a big array.
 *
 * <p>This method may be used whenever a big array range check is needed.
 *
 * @param a a big array.
 * @param offset a start index.
```

```
  * @param length a length (the number of elements in the range).
  * @throws IllegalArgumentException if {@code length} is negative.
  * @throws ArrayIndexOutOfBoundsException if {@code offset} is negative or {@code offset}+{@code length} is
greater than the big array length.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
  */
 @Deprecated
 public static KEY_GENERIC void ensureOffsetLength(final KEY_GENERIC_TYPE[][] a, final long offset, final
long length) {
  BigArrays.ensureOffsetLength(length(a), offset, length);
 }

 /** Ensures that two big arrays are of the same length.

 *
  * @param a a big array.
  * @param b another big array.
  * @throws IllegalArgumentException if the two argument arrays are not of the same length.
  * @deprecated Please use the version in {@link it.unimi.dsi.fastutil.BigArrays}.
  */
 @Deprecated
 public static KEY_GENERIC void ensureSameLength(final KEY_GENERIC_TYPE[][] a, final
KEY_GENERIC_TYPE[][] b) {
  if (length(a) != length(b)) throw new IllegalArgumentException("Array size mismatch: " + length(a) + " != " +
length(b));
 }


 /** A type-specific content-based hash strategy for big arrays. */

 private static final class BigArrayHashStrategy KEY_GENERIC implements
Hash.Strategy<KEY_GENERIC_TYPE[][]>, java.io.Serializable {
  private static final long serialVersionUID = -7046029254386353129L;

  @Override
  public int hashCode(final KEY_GENERIC_TYPE[][] o) { return java.util.Arrays.deepHashCode(o); }

  @Override
  public boolean equals(final KEY_GENERIC_TYPE[][] a, final KEY_GENERIC_TYPE[][] b) { return
BIG_ARRAYS.equals(a,
 b); }
 }

 /** A type-specific content-based hash strategy for big arrays.
  *
  * <p>This hash strategy may be used in custom hash collections whenever keys are
  * big arrays, and they must be considered equal by content. This strategy
  * will handle {@code null} correctly, and it is serializable.
```

```
 */

@SuppressWarnings({"rawtypes"})
public static final Hash.Strategy HASH_STRATEGY = new BigArrayHashStrategy();

private static final int SMALL = 7;
private static final int MEDIUM = 40;

private static KEY_GENERIC void vecSwap(final KEY_GENERIC_TYPE[][] x, long a, long b, final long n) {
 for(int i = 0; i < n; i++, a++, b++) swap(x, a, b);
}

private static KEY_GENERIC long med3(final KEY_GENERIC_TYPE x[][], final long a, final long b, final long c,
KEY_COMPARATOR KEY_GENERIC comp) {
 int ab = comp.compare(BigArrays.get(x, a), BigArrays.get(x, b));
 int ac = comp.compare(BigArrays.get(x, a), BigArrays.get(x, c));
 int bc = comp.compare(BigArrays.get(x, b), BigArrays.get(x,
c));
 return (ab < 0 ?
  (bc < 0 ? b : ac < 0 ? c : a) :
  (bc > 0 ? b : ac > 0 ? c : a));
}

private static KEY_GENERIC void selectionSort(final KEY_GENERIC_TYPE[][] a, final long from, final long to,
final KEY_COMPARATOR KEY_GENERIC comp) {
 for(long i = from; i < to - 1; i++) {
  long m = i;
  for(long j = i + 1; j < to; j++) if (comp.compare(BigArrays.get(a, j), BigArrays.get(a, m)) < 0) m = j;
  if (m != i) swap(a, i, m);
 }
}

/** Sorts the specified range of elements according to the order induced by the specified
 * comparator using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * @param x the big array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 *
 @param comp the comparator to determine the sorting order.
 */
public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[][] x, final long from, final long to,
final KEY_COMPARATOR KEY_GENERIC comp) {
 final long len = to - from;
```

```
// Selection sort on smallest arrays
if (len < SMALL) {
 selectionSort(x, from, to, comp);
 return;
}

// Choose a partition element, v
long m = from + len / 2;  // Small arrays, middle element
if (len > SMALL) {
 long l = from;
 long n = to - 1;
 if (len > MEDIUM) {  // Big arrays, pseudomedian of 9
  long s = len / 8;
  l = med3(x, l, l + s, l + 2 * s, comp);
  m = med3(x, m - s, m, m + s, comp);
  n = med3(x, n - 2 * s, n - s, n, comp);
 }
 m = med3(x, l, m, n, comp); // Mid-size, med of 3
}

final KEY_GENERIC_TYPE v = BigArrays.get(x, m);

// Establish Invariant: v* (<v)* (>v)* v*
long a = from, b = a, c = to - 1, d = c;
while(true) {
 int comparison;
 while (b <= c && (comparison = comp.compare(BigArrays.get(x,
b), v)) <= 0) {
  if (comparison == 0) swap(x, a++, b);
  b++;
 }
 while (c >= b && (comparison = comp.compare(BigArrays.get(x, c), v)) >=0) {
  if (comparison == 0) swap(x, c, d--);
  c--;
 }
 if (b > c) break;
 swap(x, b++, c--);
}

// Swap partition elements back to middle
long s, n = to;
s = Math.min(a - from, b - a);
vecSwap(x, from, b - s, s);
s = Math.min(d - c, n - d- 1);
vecSwap(x, b, n - s, s);

// Recursively sort non-partition-elements
```

```
 if ((s = b - a) > 1) quickSort(x, from, from + s, comp);
 if ((s = d - c) > 1) quickSort(x, n - s, n, comp);

}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private static KEY_GENERIC long med3(final KEY_GENERIC_TYPE x[][], final long a, final long b, final long c)
{
 int ab = KEY_CMP(BigArrays.get(x, a), BigArrays.get(x, b));
 int ac = KEY_CMP(BigArrays.get(x, a), BigArrays.get(x, c));
 int bc = KEY_CMP(BigArrays.get(x, b), BigArrays.get(x, c));
 return (ab < 0 ?
  (bc < 0 ? b : ac < 0 ? c : a)
:
  (bc > 0 ? b : ac > 0 ? c : a));
}


SUPPRESS_WARNINGS_KEY_UNCHECKED
private static KEY_GENERIC void selectionSort(final KEY_GENERIC_TYPE[][] a, final long from, final long to)
{
 for(long i = from; i < to - 1; i++) {
  long m = i;
  for(long j = i + 1; j < to; j++) if (KEY_LESS(BigArrays.get(a, j), BigArrays.get(a, m))) m = j;
  if (m != i) swap(a, i, m);
 }
}

/** Sorts the specified big array according to the order induced by the specified
 * comparator using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * @param x the big array to be sorted.
 * @param comp the comparator to determine the sorting order.
 *
 */
public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[][] x, final KEY_COMPARATOR
KEY_GENERIC comp) {
 quickSort(x, 0, BIG_ARRAYS.length(x),
comp);
}

/** Sorts the specified range of elements according to the natural ascending order using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
```

```
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * @param x the big array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[][] x, final long from, final long to) {
 final long len = to - from;

 // Selection sort on smallest arrays
 if (len < SMALL) {
  selectionSort(x, from, to);
  return;
 }

 // Choose a partition element, v
 long m = from + len / 2;  // Small arrays, middle element
 if (len > SMALL) {
  long l = from;
  long n = to - 1;
  if (len > MEDIUM) {  //
Big arrays, pseudomedian of 9
   long s = len / 8;
   l = med3(x, l, l + s, l + 2 * s);
   m = med3(x, m - s, m, m + s);
   n = med3(x, n - 2 * s, n - s, n);
  }
  m = med3(x, l, m, n); // Mid-size, med of 3
 }

 final KEY_GENERIC_TYPE v = BigArrays.get(x, m);

 // Establish Invariant: v* (<v)* (>v)* v*
 long a = from, b = a, c = to - 1, d = c;
 while(true) {
  int comparison;
  while (b <= c && (comparison = KEY_CMP(BigArrays.get(x, b), v)) <= 0) {
   if (comparison == 0) swap(x, a++, b);
   b++;
  }
  while (c >= b && (comparison = KEY_CMP(BigArrays.get(x, c), v)) >=0) {
   if (comparison == 0) swap(x, c, d--);
   c--;
  }
  if (b > c) break;
```

```
 swap(x, b++, c--);
 }

 // Swap partition elements back to middle
 long s, n = to;
 s = Math.min(a - from, b - a);
 vecSwap(x, from, b - s, s);
 s = Math.min(d - c, n - d- 1);
 vecSwap(x, b, n - s, s);

 // Recursively sort non-partition-elements
 if ((s = b - a) > 1) quickSort(x, from, from + s);
 if ((s = d -
c) > 1) quickSort(x, n - s, n);

 }


 /** Sorts the specified big array according to the natural ascending order using quicksort.
  *
  * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
  * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
  * 1249&minus;1265, 1993.
  *
  * @param x the big array to be sorted.
  */

 public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[][] x) {
  quickSort(x, 0, BIG_ARRAYS.length(x));
 }




#if ! KEY_CLASS_Boolean

 /**
  * Searches a range of the specified big array for the specified value using
  * the binary search algorithm. The range must be sorted prior to making this call.
  * If it is not sorted, the results are undefined. If the range contains multiple elements with
  * the specified value, there is no guarantee which one will be found.
  *
  * @param a the big array to be searched.
  * @param from  the index of the first element (inclusive)
 to be searched.
  * @param to  the index of the last element (exclusive) to be searched.
  * @param key the value to be searched for.
  * @return index of the search key, if it is contained in the big array;
```

```
 *          otherwise, <code>(-(<i>insertion point</i>) - 1)</code>.  The <i>insertion
 *          point</i> is defined as the the point at which the value would
 *          be inserted into the big array: the index of the first
 *          element greater than the key, or the length of the big array, if all
 *          elements in the big array are less than the specified key.  Note
 *          that this guarantees that the return value will be &gt;= 0 if
 *          and only if the key is found.
 * @see java.util.Arrays
 */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public static KEY_GENERIC long binarySearch(final KEY_GENERIC_TYPE[][] a, long from, long to, final
KEY_GENERIC_TYPE key) {
  KEY_GENERIC_TYPE midVal;
  to--;
  while (from <= to) {
   final
 long mid = (from + to) >>> 1;
   midVal = BigArrays.get(a, mid);
#if KEYS_PRIMITIVE
   if (midVal < key) from = mid + 1;
   else if (midVal > key) to = mid - 1;
   else return mid;
#else
   final int cmp = ((Comparable KEY_SUPER_GENERIC)midVal).compareTo(key);
   if (cmp < 0) from = mid + 1;
   else if (cmp > 0) to = mid - 1;
   else return mid;
#endif
  }
  return -(from + 1);
 }

 /**
 * Searches a big array for the specified value using
 * the binary search algorithm. The range must be sorted prior to making this call.
 * If it is not sorted, the results are undefined. If the range contains multiple elements with
 * the specified value, there is no guarantee which one will be found.
 *
 * @param a the big array to be searched.
 * @param key the value to be searched for.
 * @return index of the search key, if it is contained in the big array;
 *          otherwise, <code>(-(<i>insertion point</i>) - 1)</code>.  The <i>insertion
 *          point</i> is defined as
 the the point at which the value would
 *          be inserted into the big array: the index of the first
 *          element greater than the key, or the length of the big array, if all
 *          elements in the big array are less than the specified key.  Note
 *          that this guarantees that the return value will be &gt;= 0 if
```

```
 *            and only if the key is found.
 * @see java.util.Arrays
 */
public static KEY_GENERIC long binarySearch(final KEY_GENERIC_TYPE[][] a, final KEY_TYPE key) {
 return binarySearch(a, 0, BIG_ARRAYS.length(a), key);
}

/**
 * Searches a range of the specified big array for the specified value using
 * the binary search algorithm and a specified comparator. The range must be sorted following the comparator prior
to making this call.
 * If it is not sorted, the results are undefined. If the range contains multiple elements with
 * the specified value, there is no guarantee which one will be found.
 *
 * @param a the big
array to be searched.
 * @param from  the index of the first element (inclusive) to be searched.
 * @param to  the index of the last element (exclusive) to be searched.
 * @param key the value to be searched for.
 * @param c a comparator.
 * @return index of the search key, if it is contained in the big array;
 *            otherwise, <code>(-(<i>insertion point</i>) - 1)</code>.  The <i>insertion
 *            point</i> is defined as the the point at which the value would
 *            be inserted into the big array: the index of the first
 *            element greater than the key, or the length of the big array, if all
 *            elements in the big array are less than the specified key.  Note
 *            that this guarantees that the return value will be &gt;= 0 if
 *            and only if the key is found.
 * @see java.util.Arrays
 */
public static KEY_GENERIC long binarySearch(final KEY_GENERIC_TYPE[][] a, long from, long to, final
KEY_GENERIC_TYPE key,
final KEY_COMPARATOR KEY_GENERIC c) {
 KEY_GENERIC_TYPE midVal;
 to--;
 while (from <= to) {
  final long mid = (from + to) >>> 1;
  midVal = BigArrays.get(a, mid);
  final int cmp = c.compare(midVal, key);
  if (cmp < 0) from = mid + 1;
  else if (cmp > 0) to = mid - 1;
  else return mid; // key found
 }
 return -(from + 1);
}

/**
 * Searches a big array for the specified value using
```

```
  * the binary search algorithm and a specified comparator. The range must be sorted following the comparator prior
to making this call.
  * If it is not sorted, the results are undefined. If the range contains multiple elements with
  * the specified value, there is no guarantee which one will be found.
  *
  * @param a the big array to be searched.
  * @param key the value to be searched for.
  * @param c a comparator.
  * @return index of the search key, if it is contained in the big array;
  *          otherwise, <code>(-(<i>insertion point</i>) - 1)</code>.  The <i>insertion
  *
          point</i> is defined as the the point at which the value would
  *          be inserted into the big array: the index of the first
  *          element greater than the key, or the length of the big array, if all
  *          elements in the big array are less than the specified key.  Note
  *          that this guarantees that the return value will be &gt;= 0 if
  *          and only if the key is found.
  * @see java.util.Arrays
  */
 public static KEY_GENERIC long binarySearch(final KEY_GENERIC_TYPE[][] a, final KEY_GENERIC_TYPE
key, final KEY_COMPARATOR KEY_GENERIC c) {
  return binarySearch(a, 0, BIG_ARRAYS.length(a), key, c);
 }


#if KEYS_PRIMITIVE
 /** The size of a digit used during radix sort (must be a power of 2). */
 private static final int DIGIT_BITS = 8;
 /** The mask to extract a digit of {@link #DIGIT_BITS} bits. */
 private static final int DIGIT_MASK = (1 << DIGIT_BITS) - 1;
 /** The number of digits per element. */
 private static final
 int DIGITS_PER_ELEMENT = KEY_CLASS.SIZE / DIGIT_BITS;

 /** This method fixes negative numbers so that the combination exponent/significand is lexicographically sorted. */
#if KEY_CLASS_Double
 private static final long fixDouble(final double d) {
  final long l = Double.doubleToRawLongBits(d);
  return l >= 0 ? l : l ^ 0x7FFFFFFFFFFFFFFFL;
 }
#elif KEY_CLASS_Float
 private static final int fixFloat(final float f) {
  final int i = Float.floatToRawIntBits(f);
  return i >= 0 ? i : i ^ 0x7FFFFFFF;
 }
#endif
```

```
/** Sorts the specified big array using radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993),
 * and further improved using the digit-oracle idea described by
 * Juha K&auml;rkk&auml;inen and Tommi Rantala in &ldquo;Engineering radix sort for strings&rdquo;,
 * <i>String Processing and Information Retrieval,
15th International Symposium</i>, volume 5280 of
 * Lecture Notes in Computer Science, pages 3&minus;14, Springer (2008).
 *
 * <p>This implementation is significantly faster than quicksort
 * already at small sizes (say, more than 10000 elements), but it can only
 * sort in ascending order.
 * It will allocate a support array of bytes with the same number of elements as the array to be sorted.
 *
 * @param a the big array to be sorted.
 */
public static void radixSort(final KEY_TYPE[][] a) {
 radixSort(a, 0, BIG_ARRAYS.length(a));
}

/** Sorts the specified big array using radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993),
 * and further improved using the digit-oracle idea described by
 * Juha K&auml;rkk&auml;inen and Tommi Rantala in &ldquo;Engineering radix sort for strings&rdquo;,

* <i>String Processing and Information Retrieval, 15th International Symposium</i>, volume 5280 of
 * Lecture Notes in Computer Science, pages 3&minus;14, Springer (2008).
 *
 * <p>This implementation is significantly faster than quicksort
 * already at small sizes (say, more than 10000 elements), but it can only
 * sort in ascending order.
 * It will allocate a support array of bytes with the same number of elements as the array to be sorted.
 *
 * @param a the big array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
public static void radixSort(final KEY_TYPE[][] a, final long from, final long to) {
 final int maxLevel = DIGITS_PER_ELEMENT - 1;

 final int stackSize = ((1 << DIGIT_BITS) - 1) * (DIGITS_PER_ELEMENT - 1) + 1;
 final long[] offsetStack = new long[stackSize];
 int offsetPos = 0;
 final long[] lengthStack = new long[stackSize];
 int lengthPos
```

```
 = 0;
  final int[] levelStack = new int[stackSize];
  int levelPos = 0;

  offsetStack[offsetPos++] = from;
  lengthStack[lengthPos++] = to - from;
  levelStack[levelPos++] = 0;

  final long[] count = new long[1 << DIGIT_BITS];
  final long[] pos = new long[1 << DIGIT_BITS];
  final byte[][] digit = ByteBigArrays.newBigArray(to - from);

  while(offsetPos > 0) {
   final long first = offsetStack[--offsetPos];
   final long length = lengthStack[--lengthPos];
   final int level = levelStack[--levelPos];
#if KEY_CLASS_Character
   final int signMask = 0;
#else
   final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
#endif

   if (length < MEDIUM) {
    selectionSort(a, first, first + length);
    continue;
   }

   final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
shift that extract the right byte from a key

   // Count keys.

   for(long i = length; i-- != 0;) BigArrays.set(digit, i, (byte)(((KEY2LEXINT(BigArrays.get(a,
 first + i)) >>> shift) & DIGIT_MASK) ^ signMask));
   for(long i = length; i-- != 0;) count[BigArrays.get(digit, i) & 0xFF]++;
   // Compute cumulative distribution and push non-singleton keys on stack.
   int lastUsed = -1;

   long p = 0;
   for(int i = 0; i < 1 << DIGIT_BITS; i++) {
    if (count[i] != 0) {
     lastUsed = i;
     if (level < maxLevel && count[i] > 1){
      //System.err.println(" Pushing " + new StackEntry(first + pos[i - 1], first + pos[i], level + 1));
      offsetStack[offsetPos++] = p + first;
      lengthStack[lengthPos++] = count[i];
      levelStack[levelPos++] = level + 1;
     }
```

```
    }
    pos[i] = (p += count[i]);
   }

   // When all slots are OK, the last slot is necessarily OK.
   final long end = length - count[lastUsed];
   count[lastUsed] = 0;

   // i moves through the start of each block
   int c = -1;
   for(long i = 0, d; i < end; i += count[c], count[c] = 0) {
    KEY_TYPE t = BigArrays.get(a, i +first);
    c = BigArrays.get(digit,
 i) & 0xFF;
    while((d = --pos[c]) > i) {
     final KEY_TYPE z = t;
     final int zz = c;
     t = BigArrays.get(a, d + first);
     c = BigArrays.get(digit, d) & 0xFF;
     BigArrays.set(a, d + first, z);
     BigArrays.set(digit, d, (byte)zz);
    }

    BigArrays.set(a, i + first, t);
   }
  }
 }


 private static void selectionSort(final KEY_TYPE[][] a, final KEY_TYPE[][] b, final long from, final long to) {
  for(long i = from; i < to - 1; i++) {
   long m = i;
   for(long j = i + 1; j < to; j++)
    if (KEY_LESS(BigArrays.get(a, j), BigArrays.get(a, m)) || KEY_CMP_EQ(BigArrays.get(a, j), BigArrays.get(a,
m)) && KEY_LESS(BigArrays.get(b, j), BigArrays.get(b, m))) m = j;

   if (m != i) {
    KEY_TYPE t = BigArrays.get(a, i);
    BigArrays.set(a, i, BigArrays.get(a, m));
    BigArrays.set(a, m, t);
    t = BigArrays.get(b, i);
    BigArrays.set(b, i, BigArrays.get(b, m));
    BigArrays.set(b, m, t);
   }
  }
 }

 /** Sorts the specified pair of big arrays lexicographically using
```

```
radix sort.
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993),
 * and further improved using the digit-oracle idea described by
 * Juha K&auml;rkk&auml;inen and Tommi Rantala in &ldquo;Engineering radix sort for strings&rdquo;,
 * <i>String Processing and Information Retrieval, 15th International Symposium</i>, volume 5280 of
 * Lecture Notes in Computer Science, pages 3&minus;14, Springer (2008).
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of elements
 * in the same position in the two provided arrays will be considered a single key, and permuted
 * accordingly. In the end, either {@code a[i] &lt; a[i + 1]} or {@code a[i] == a[i + 1]} and {@code b[i] &lt;= b[i +
1]}.
 *
 * <p>This implementation is significantly faster than quicksort
 * already at small sizes
(say, more than 10000 elements), but it can only
 * sort in ascending order. It will allocate a support array of bytes with the same number of elements as the arrays to
be sorted.
 *
 * @param a the first big array to be sorted.
 * @param b the second big array to be sorted.
 */

public static void radixSort(final KEY_TYPE[][] a, final KEY_TYPE[][] b) {
 radixSort(a, b, 0, BIG_ARRAYS.length(a));
}

/** Sorts the specified pair of big arrays lexicographically using radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993),
 * and further improved using the digit-oracle idea described by
 * Juha K&auml;rkk&auml;inen and Tommi Rantala in &ldquo;Engineering radix sort for strings&rdquo;,
 * <i>String Processing and Information Retrieval, 15th International Symposium</i>, volume 5280 of
 * Lecture Notes in
Computer Science, pages 3&minus;14, Springer (2008).
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of elements
 * in the same position in the two provided arrays will be considered a single key, and permuted
 * accordingly. In the end, either {@code a[i] &lt; a[i + 1]} or {@code a[i] == a[i + 1]} and {@code b[i] &lt;= b[i +
1]}.
 *
 * <p>This implementation is significantly faster than quicksort
 * already at small sizes (say, more than 10000 elements), but it can only
 * sort in ascending order. It will allocate a support array of bytes with the same number of elements as the arrays to
be sorted.
 *
 * @param a the first big array to be sorted.
```

```
  * @param b the second big array to be sorted.
  * @param from the index of the first element (inclusive) to be sorted.
  * @param to the index of the last element (exclusive) to be sorted.
  */
 public static void radixSort(final KEY_TYPE[][] a, final KEY_TYPE[][] b, final long from, final
 long to) {
  final int layers = 2;
  if (BIG_ARRAYS.length(a) != BIG_ARRAYS.length(b)) throw new IllegalArgumentException("Array size
mismatch.");
  final int maxLevel = DIGITS_PER_ELEMENT * layers - 1;

  final int stackSize = ((1 << DIGIT_BITS) - 1) * (layers * DIGITS_PER_ELEMENT - 1) + 1;
  final long[] offsetStack = new long[stackSize];
  int offsetPos = 0;
  final long[] lengthStack = new long[stackSize];
  int lengthPos = 0;
  final int[] levelStack = new int[stackSize];
  int levelPos = 0;

  offsetStack[offsetPos++] = from;
  lengthStack[lengthPos++] = to - from;
  levelStack[levelPos++] = 0;

  final long[] count = new long[1 << DIGIT_BITS];
  final long[] pos = new long[1 << DIGIT_BITS];
  final byte[][] digit = ByteBigArrays.newBigArray(to - from);

  while(offsetPos > 0) {
   final long first = offsetStack[--offsetPos];
   final long length = lengthStack[--lengthPos];
   final int level = levelStack[--levelPos];
#if KEY_CLASS_Character
   final int signMask = 0;
#else
   final
 int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
#endif

   if (length < MEDIUM) {
    selectionSort(a, b, first, first + length);
    continue;
   }

   final KEY_TYPE[][] k = level < DIGITS_PER_ELEMENT ? a : b; // This is the key array
   final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
shift that extract the right byte from a key

   // Count keys.
```

```
  for(long i = length; i-- != 0;) BigArrays.set(digit, i, (byte)(((KEY2LEXINT(BigArrays.get(k, first + i)) >>> shift) &
DIGIT_MASK) ^ signMask));
  for(long i = length; i-- != 0;) count[BigArrays.get(digit, i) & 0xFF]++;
  // Compute cumulative distribution and push non-singleton keys on stack.
  int lastUsed = -1;

  long p = 0;
  for(int i = 0; i < 1 << DIGIT_BITS; i++) {
   if (count[i] != 0) {
    lastUsed = i;
    if (level < maxLevel && count[i] > 1){
     offsetStack[offsetPos++] = p + first;
     lengthStack[lengthPos++] = count[i];
     levelStack[levelPos++]
 = level + 1;
    }
   }
   pos[i] = (p += count[i]);
  }

  // When all slots are OK, the last slot is necessarily OK.
  final long end = length - count[lastUsed];
  count[lastUsed] = 0;

  // i moves through the start of each block
  int c = -1;
  for(long i = 0, d; i < end; i += count[c], count[c] = 0) {
   KEY_TYPE t = BigArrays.get(a, i + first);
   KEY_TYPE u = BigArrays.get(b, i + first);
   c = BigArrays.get(digit, i) & 0xFF;
   while((d = --pos[c]) > i) {
    KEY_TYPE z = t;
    final int zz = c;
    t = BigArrays.get(a, d + first);
    BigArrays.set(a, d + first, z);
    z = u;
    u = BigArrays.get(b, d + first);
    BigArrays.set(b, d + first, z);
    c = BigArrays.get(digit, d) & 0xFF;
    BigArrays.set(digit, d, (byte)zz);
   }

   BigArrays.set(a, i + first, t);
   BigArrays.set(b, i + first, u);
  }
 }
}
```

```
 private static final int RADIXSORT_NO_REC = 1024;

 private static KEY_GENERIC void insertionSortIndirect(final
 long[][] perm, final KEY_TYPE[][] a, final KEY_TYPE[][] b, final long from, final long to) {
  for (long i = from; ++i < to;) {
   long t = BigArrays.get(perm, i);
   long j = i;
   for (long u = BigArrays.get(perm, j - 1); KEY_LESS(BigArrays.get(a, t), BigArrays.get(a, u)) ||
KEY_CMP_EQ(BigArrays.get(a, t), BigArrays.get(a, u)) && KEY_LESS(BigArrays.get(b, t), BigArrays.get(b, u));
u = BigArrays.get(perm, --j - 1)) {
    BigArrays.set(perm, j, u);
    if (from == j - 1) {
     --j;
     break;
    }
   }
   BigArrays.set(perm, j, t);
  }
 }

 /** Sorts the specified pair of arrays lexicographically using indirect radix sort.
  *
  * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
  * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
  *
  * <p>This method implement an <em>indirect</em> sort. The elements of {@code perm} (which must
  * be exactly the numbers in the interval
 {@code [0..length(perm))}) will be permuted so that
  * {@code a[perm[i]] &le; a[perm[i + 1]]} or {@code a[perm[i]] == a[perm[i + 1]]} and {@code b[perm[i]] &le;
b[perm[i + 1]]}.
  *
  * <p>This implementation will allocate, in the stable case, a further support array as large as {@code perm} (note
that the stable
  * version is slightly faster).
  *
  * @param perm a permutation array indexing {@code a}.
  * @param a the array to be sorted.
  * @param b the second array to be sorted.
  * @param stable whether the sorting algorithm should be stable.
  */
 public static void radixSortIndirect(final long[][] perm, final KEY_TYPE[][] a, final KEY_TYPE[][] b, final
boolean stable) {
  ensureSameLength(a, b);
  radixSortIndirect(perm, a, b, 0, BigArrays.length(a), stable);
 }

 /** Sorts the specified pair of arrays lexicographically using indirect radix sort.
```

```
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering
radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implement an <em>indirect</em> sort. The elements of {@code perm} (which must
 * be exactly the numbers in the interval {@code [0..length(perm))}) will be permuted so that
 * {@code a[perm[i]] &le; a[perm[i + 1]]} or {@code a[perm[i]] == a[perm[i + 1]]} and {@code b[perm[i]] &le;
b[perm[i + 1]]}.
 *
 * <p>This implementation will allocate, in the stable case, a further support array as large as {@code perm} (note
that the stable
 * version is slightly faster).
 *
 * @param perm a permutation array indexing {@code a}.
 * @param a the array to be sorted.
 * @param b the second array to be sorted.
 * @param from the index of the first element of {@code perm} (inclusive) to be permuted.
 * @param to the index of the last element of {@code perm} (exclusive) to be permuted.
 * @param stable whether the sorting algorithm should be stable.
 */
public static void radixSortIndirect(final
long[][] perm, final KEY_TYPE[][] a, final KEY_TYPE[][] b, final long from, final long to, final boolean stable) {
 if (to - from < RADIXSORT_NO_REC) {
  insertionSortIndirect(perm, a, b, from, to);
  return;
 }

 final int layers = 2;
 final int maxLevel = DIGITS_PER_ELEMENT * layers - 1;

 final int stackSize = ((1 << DIGIT_BITS) - 1) * (layers * DIGITS_PER_ELEMENT - 1) + 1;
 int stackPos = 0;
 final long[] offsetStack = new long[stackSize];
 final long[] lengthStack = new long[stackSize];
 final int[] levelStack = new int[stackSize];

 offsetStack[stackPos] = from;
 lengthStack[stackPos] = to - from;
 levelStack[stackPos++] = 0;

 final long[] count = new long[1 << DIGIT_BITS];
 final long[] pos = new long[1 << DIGIT_BITS];
 final long[][] support = stable ? it.unimi.dsi.fastutil.longs.LongBigArrays.newBigArray(BigArrays.length(perm)) :
null;

 while(stackPos > 0) {
  final long first = offsetStack[--stackPos];
```

```
   final long length = lengthStack[stackPos];
   final
 int level = levelStack[stackPos];
#if KEY_CLASS_Character
   final int signMask = 0;
#else
   final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
#endif

   final KEY_TYPE[][] k = level < DIGITS_PER_ELEMENT ? a : b; // This is the key array
   final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
shift that extract the right byte from a key

   // Count keys.
   for(long i = first + length; i-- != first;) count[INT(KEY2LEXINT(BigArrays.get(k, BigArrays.get(perm, i))) >>>
shift & DIGIT_MASK ^ signMask)]++;
   // Compute cumulative distribution
   int lastUsed = -1;
   long p = stable ? 0 : first;
   for (int i = 0; i < 1 << DIGIT_BITS; i++) {
    if (count[i] != 0) lastUsed = i;
    pos[i] = (p += count[i]);
   }

   if (stable) {
    for(long i = first + length; i-- != first;) BigArrays.set(support, --pos[INT(KEY2LEXINT(BigArrays.get(k,
BigArrays.get(perm, i))) >>> shift & DIGIT_MASK ^ signMask)],
 BigArrays.get(perm, i));
    BigArrays.copy(support, 0, perm, first, length);
    p = first;
    for(int i = 0; i < 1 << DIGIT_BITS; i++) {
     if (level < maxLevel && count[i] > 1) {
      if (count[i] < RADIXSORT_NO_REC) insertionSortIndirect(perm, a, b, p, p + count[i]);
      else {
       offsetStack[stackPos] = p;
       lengthStack[stackPos] = count[i];
       levelStack[stackPos++] = level + 1;
      }
     }
     p += count[i];
    }
    java.util.Arrays.fill(count, 0);
   }
   else {
    final long end = first + length - count[lastUsed];
    // i moves through the start of each block
    int c = -1;
    for(long i = first, d; i <= end; i += count[c], count[c] = 0) {
```

```
    long t = BigArrays.get(perm, i);
    c = INT(KEY2LEXINT(BigArrays.get(k, t)) >>> shift & DIGIT_MASK ^ signMask);

    if (i < end) { // When all slots are OK, the last slot is necessarily OK.
     while((d = --pos[c]) > i) {
      final long z = t;
      t = BigArrays.get(perm, d);
      BigArrays.set(perm,
 d, z);
      c = INT(KEY2LEXINT(BigArrays.get(k, t)) >>> shift & DIGIT_MASK ^ signMask);
     }
     BigArrays.set(perm, i, t);
    }


    if (level < maxLevel && count[c] > 1) {
     if (count[c] < RADIXSORT_NO_REC) insertionSortIndirect(perm, a, b, i, i + count[c]);
     else {
      offsetStack[stackPos] = i;
      lengthStack[stackPos] = count[c];
      levelStack[stackPos++] = level + 1;
     }
    }
   }
  }
 }
}




#endif

#endif


 /** Shuffles the specified big array fragment using the specified pseudorandom number generator.
  *
  * @param a the big array to be shuffled.
  * @param from the index of the first element (inclusive) to be shuffled.
  * @param to the index of the last element (exclusive) to be shuffled.
  * @param random a pseudorandom number generator.
  * @return {@code a}.
  */
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] shuffle(final KEY_GENERIC_TYPE[][] a, final long
from, final long to, final Random random) {
  return BigArrays.shuffle(a,
```

```
 from, to, random);
 }

 /** Shuffles the specified big array using the specified pseudorandom number generator.
  *
  * @param a the big array to be shuffled.
  * @param random a pseudorandom number generator.
  * @return {@code a}.
  */
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] shuffle(final KEY_GENERIC_TYPE[][] a, final Random
random) {
  return BigArrays.shuffle(a, random);
 }


#if KEY_CLASS_Integer
#ifdef TEST

 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif KEY_CLASS_Object
  return Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }

 private static Object[] k, v, nk;
 private static KEY_TYPE kt[];
 private static KEY_TYPE nkt[];
 private static BIG_ARRAY_BIG_LIST topList;

 protected
 static void speedTest(int n, boolean b) {}

 protected static void runTest(int n) {
  KEY_TYPE[][] a = BIG_ARRAYS.newBigArray(n);
  for(int i = 0; i < n; i++) BigArrays.set(a, i, i);
  BIG_ARRAYS.copy(a, 0, a, 1, n - 2);
  assert a[0][0] == 0;
  for(int i = 0; i < n - 2; i++) assert BigArrays.get(a, i + 1) == i;
```

```
  for(int i = 0; i < n; i++) BigArrays.set(a, i, i);
  BIG_ARRAYS.copy(a, 1, a, 0, n - 1);
  for(int i = 0; i < n - 1; i++) assert BigArrays.get(a, i) == i + 1;

  for(int i = 0; i < n; i++) BigArrays.set(a, i, i);
  KEY_TYPE[] b = new KEY_TYPE[n];
  for(int i = 0; i < n; i++) b[i] = i;

  assert equals(wrap(b), a);

  System.out.println("Test OK");
  return;

 }


 public static void main(String args[]) {
  int n  = Integer.parseInt(args[1]);
  if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

  try {
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
   else if ("test".equals(args[0]))
 runTest(n);
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed: " + seed);
  }
 }

#endif
#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/BigArrays.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
```

```
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

#if KEY_CLASS_Object
import java.util.Comparator;
#endif

import it.unimi.dsi.fastutil.ints.IntArrays;

import java.util.Arrays;
import java.util.NoSuchElementException;

/** A type-specific heap-based indirect priority queue.
*
* <p>Instances of this class use an additional <em>inversion
 array</em>, of the same length of the reference array,
* to keep track of the heap position containing a given element of the reference array. The priority queue is
* represented using a heap. The heap is enlarged as needed, but it is never
* shrunk. Use the {@link #trim()} method to reduce its size, if necessary.
*
* <p>This implementation does <em>not</em> allow one to enqueue several times the same index.
*/

public class HEAP_INDIRECT_PRIORITY_QUEUE KEY_GENERIC extends
HEAP_SEMI_INDIRECT_PRIORITY_QUEUE KEY_GENERIC {

 /** The inversion array. */
 protected final int inv[];

 /** Creates a new empty queue with a given capacity and comparator.
  *
  * @param refArray the reference array.
  * @param capacity the initial capacity of this queue.
  * @param c the comparator used in this queue, or {@code null} for the natural order.
  */
 public HEAP_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, int capacity,
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
  super(refArray, capacity,
 c);
  if (capacity > 0) this.heap = new int[capacity];
```

```
 this.c = c;

 this.inv = new int[refArray.length];
 Arrays.fill(inv, -1);
}

/** Creates a new empty queue with a given capacity and using the natural order.
 *
 * @param refArray the reference array.
 * @param capacity the initial capacity of this queue.
 */
public HEAP_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, int capacity) {
 this(refArray, capacity, null);
}

/** Creates a new empty queue with capacity equal to the length of the reference array and a given comparator.
 *
 * @param refArray the reference array.
 * @param c the comparator used in this queue, or {@code null} for the natural order.
 */
public HEAP_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, KEY_COMPARATOR
KEY_SUPER_GENERIC c) {
 this(refArray, refArray.length, c);
}

/** Creates a new empty queue with capacity equal to the length of the reference array and using the natural order.
 * @param refArray the reference array.

*/
public HEAP_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray) {
 this(refArray, refArray.length, null);
}


/** Wraps a given array in a queue using a given comparator.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The first {@code size} element of the array will be rearranged so to form a heap (this is
 * more efficient than enqueing the elements of {@code a} one by one).
 *
 * @param refArray the reference array.
 * @param a an array of indices into {@code refArray}.
 * @param size the number of elements to be included in the queue.
 * @param c the comparator used in this queue, or {@code null} for the natural order.
 */
public HEAP_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, final int
size, final KEY_COMPARATOR KEY_SUPER_GENERIC c) {
```

```
 this(refArray, 0, c);
 this.heap = a;
 this.size = size;
 int i = size;
 while(i-- != 0) {
  if (inv[a[i]] != -1) throw new IllegalArgumentException("Index
" + a[i] + " appears twice in the heap");
  inv[a[i]] = i;
 }
 INDIRECT_HEAPS.makeHeap(refArray, a, inv, size, c);
}


/** Wraps a given array in a queue using a given comparator.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The elements of the array will be rearranged so to form a heap (this is
 * more efficient than enqueing the elements of {@code a} one by one).
 *
 * @param refArray the reference array.
 * @param a an array of indices into {@code refArray}.
 * @param c the comparator used in this queue, or {@code null} for the natural order.
 */
 public HEAP_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, final
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
 this(refArray, a, a.length, c);
}

/** Wraps a given array in a queue using the natural order.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The first {@code size} element of the array will be rearranged so to form a heap
(this is
 * more efficient than enqueing the elements of {@code a} one by one).
 *
 * @param refArray the reference array.
 * @param a an array of indices into {@code refArray}.
 * @param size the number of elements to be included in the queue.
 */
 public HEAP_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, int size) {
 this(refArray, a, size, null);
}


/** Wraps a given array in a queue using the natural order.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The elements of the array will be rearranged so to form a heap (this is
```

```
 * more efficient than enqueing the elements of {@code a} one by one).
 *
 * @param refArray the reference array.
 * @param a an array of indices into {@code refArray}.
 */
public HEAP_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a) {
 this(refArray, a, a.length);
}

@Override
public void enqueue(final int x) {
 if (inv[x] >= 0) throw new IllegalArgumentException("Index
" + x + " belongs to the queue");

 if (size == heap.length) heap = IntArrays.grow(heap, size + 1);

 inv[heap[size] = x] = size++;

 INDIRECT_HEAPS.upHeap(refArray, heap, inv, size, size - 1, c);
}

@Override
public boolean contains(final int index) {
 return inv[index] >= 0;
}

@Override
public int dequeue() {
 if (size == 0) throw new NoSuchElementException();
 final int result = heap[0];
 if (--size != 0) inv[heap[0] = heap[size]] = 0;
 inv[result] = -1;

 if (size != 0) INDIRECT_HEAPS.downHeap(refArray, heap, inv, size, 0, c);
 return result;
}

@Override
public void changed() {
 INDIRECT_HEAPS.downHeap(refArray, heap, inv, size, 0, c);
}

@Override
public void changed(final int index) {
 final int pos = inv[index];
 if (pos < 0) throw new IllegalArgumentException("Index " + index + " does not belong to the queue");
 final int newPos = INDIRECT_HEAPS.upHeap(refArray, heap, inv, size, pos, c);
 INDIRECT_HEAPS.downHeap(refArray, heap,
```

```
inv, size, newPos, c);
}

/** Rebuilds this queue in a bottom-up fashion (in linear time). */
@Override
public void allChanged() { INDIRECT_HEAPS.makeHeap(refArray, heap, inv, size, c); }

@Override
public boolean remove(final int index) {
 final int result = inv[index];
 if (result < 0) return false;
 inv[index] = -1;

 if (result < --size) {
  inv[heap[result] = heap[size]] = result;
  final int newPos = INDIRECT_HEAPS.upHeap(refArray, heap, inv, size, result, c);
  INDIRECT_HEAPS.downHeap(refArray, heap, inv, size, newPos, c);
 }

 return true;
}

@Override
public void clear() {
 size = 0;
 Arrays.fill(inv, -1);
}

#ifdef TEST

/** The original class, now just used for testing. */

private static class TestQueue {

 /** The reference array */
 private KEY_TYPE refArray[];
 /** Its length */
 private int N;
 /** The number of elements in the heaps */
 private int n;
 /** The two comparators */
 private KEY_COMPARATOR primaryComp, secondaryComp;
 /**
Two indirect heaps are used, called {@code primary} and {@code secondary}. Each of them contains
  a permutation of {@code n} among the indices 0, 1, ..., {@code N}-1 in such a way that the corresponding
  objects be sorted with respect to the two comparators.
  We also need an array {@code inSec[]} so that {@code inSec[k]} is the index of {@code secondary}
  containing {@code k}.
```

```
  */
  private int primary[], secondary[], inSec[];

  /** Builds a double indirect priority queue.
   *  @param refArray The reference array.
   *  @param primaryComp The primary comparator.
   *  @param secondaryComp The secondary comparator.
   */
  public TestQueue(KEY_TYPE refArray[], KEY_COMPARATOR primaryComp, KEY_COMPARATOR
secondaryComp) {
   this.refArray = refArray;
   this.N = refArray.length;
   assert this.N != 0;
   this.n = 0;
   this.primaryComp = primaryComp;
   this.secondaryComp = secondaryComp;
   this.primary = new int[N];
   this.secondary = new int[N];
   this.inSec = new int[N];
   java.util.Arrays.fill(inSec,
 -1);
  }

  /** Adds an index to the queue. Notice that the index should not be already present in the queue.
   *  @param i The index to be added
   */
  public void add(int i) {
   if (i < 0 || i >= refArray.length) throw new IndexOutOfBoundsException();
   if (inSec[i] >= 0) throw new IllegalArgumentException();
   primary[n] = i;
   secondary[n] = i; inSec[i] = n;
   n++;
   swimPrimary(n-1);
   swimSecondary(n-1);
  }

  /** Heapify the primary heap.
   *  @param i The index of the heap to be heapified.
   */
  private void heapifyPrimary(int i) {
   int dep = primary[i];
   int child;

   while ((child = 2*i+1) < n) {
    if (child+1 < n && primaryComp.compare(refArray[primary[child+1]], refArray[primary[child]]) < 0) child++;
    if (primaryComp.compare(refArray[dep], refArray[primary[child]]) <= 0) break;
    primary[i] = primary[child];
    i = child;
```

```
  }
  primary[i] = dep;
 }

 /** Heapify the secondary heap.
  *  @param i The index of the
heap to be heapified.
  */
 private void heapifySecondary(int i) {
  int dep = secondary[i];
  int child;

  while ((child = 2*i+1) < n) {
   if (child+1 < n && secondaryComp.compare(refArray[secondary[child+1]], refArray[secondary[child]]) < 0)
child++;
   if (secondaryComp.compare(refArray[dep], refArray[secondary[child]]) <= 0) break;
   secondary[i] = secondary[child]; inSec[secondary[i]] = i;
   i = child;
  }
  secondary[i] = dep; inSec[secondary[i]] = i;
 }

 /** Swim and heapify the primary heap.
  *  @param i The index to be moved.
  */
 private void swimPrimary(int i) {
  int dep = primary[i];
  int parent;

  while (i != 0 && (parent = (i - 1) / 2) >= 0) {
   if (primaryComp.compare(refArray[primary[parent]], refArray[dep]) <= 0) break;
   primary[i] = primary[parent];
   i = parent;
  }
  primary[i] = dep;
  heapifyPrimary(i);
 }

 /** Swim and heapify the secondary heap.
  *  @param i The index to be moved.
  */
 private void swimSecondary(int
i) {
  int dep = secondary[i];
  int parent;

  while (i != 0 && (parent = (i - 1) / 2) >= 0) {
   if (secondaryComp.compare(refArray[secondary[parent]], refArray[dep]) <= 0) break;
```

```
  secondary[i] = secondary[parent]; inSec[secondary[i]] = i;
  i = parent;
  }
 secondary[i] = dep; inSec[secondary[i]] = i;
 heapifySecondary(i);
}

/** Returns the minimum element with respect to the primary comparator.
 @return the minimum element.
*/
public int top() {
 if (n == 0) throw new java.util.NoSuchElementException();
 return primary[0];
}

/** Returns the minimum element with respect to the secondary comparator.
 @return the minimum element.
*/
public int secTop() {
 if (n == 0) throw new java.util.NoSuchElementException();
 return secondary[0];
}

/** Removes the minimum element with respect to the primary comparator.
 *  @return the removed element.
 */
public boolean remove() {
 if (n == 0) throw new java.util.NoSuchElementException();
 if
(inSec[primary[0]] == -1) return false;
 int result = primary[0];
 int ins = inSec[result];
 inSec[result] = -1;
 // Copy a leaf
 primary[0] = primary[n-1];
 if (ins == n-1) {
  n--;
  heapifyPrimary(0);
  return true;
  }
 secondary[ins] = secondary[n-1];
 inSec[secondary[ins]] = ins;
 // Heapify
 n--;
 heapifyPrimary(0);
 swimSecondary(ins);
 return true;
}
```

```
public void clear() {
 while(size() != 0) remove();
}

public void remove(int index) {
 if (index >= refArray.length) throw new IndexOutOfBoundsException();
 if (inSec[index] == -1) return;
 int ins = inSec[index];
 inSec[index] = -1;
 // Copy a leaf
 primary[ins] = primary[n-1];
 if (ins == n-1) {
  n--;
  swimPrimary(ins);
  return;
 }
 secondary[ins] = secondary[n-1];
 inSec[secondary[ins]] = ins;
 // Heapify
 n--;
 swimPrimary(ins);
 swimSecondary(ins);
}

/** Signals that the minimum element with respect to the comparator has changed.
 */
public
void change() {
 if (n == 0) throw new java.util.NoSuchElementException();
 if (inSec[primary[0]] == -1) throw new IllegalArgumentException();
 int ins = inSec[primary[0]];
 heapifyPrimary(0);
 swimSecondary(ins);
}

public void change(int index) {
 if (index >= refArray.length) throw new IndexOutOfBoundsException();
 if (inSec[index] == -1) throw new IllegalArgumentException();
 if (n == 0) throw new java.util.NoSuchElementException();
 int ins = inSec[index];
 swimPrimary(ins);
 swimSecondary(ins);
}

/** Returns the number of elements in the queue.
 *  @return the size of the queue
 */
```

```
 public int size() {
  return n;
 }



 public String toString() {
  String s = "[";
  for (int i = 0; i < n; i++)
   s += refArray[primary[i]]+", ";
  return s+ "]";
 }
}

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte
|| KEY_CLASS_Short || KEY_CLASS_Character
 return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
 return r.NEXT_KEY();
#elif KEY_CLASS_Object
 return Integer.toBinaryString(r.nextInt());
#else
 return new java.io.Serializable() {};
#endif
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition p = new java.text.FieldPosition(0);

private static String format(double d) {
 StringBuffer s = new StringBuffer();
 return format.format(d, s, p).toString();
}

private static void speedTest(int n, boolean comp) {
 System.out.println("There are presently no speed tests for this class.");
}


private static void fatal(String msg) {
 System.out.println(msg);
 System.exit(1);
}
```

```
 private static void ensure(boolean cond, String msg) {
  if (cond) return;
  fatal(msg);
 }

 private static boolean heapEqual(int[] a, int[] b, int sizea, int sizeb) {
  if (sizea != sizeb) return false;
  while(sizea-- != 0) if (a[sizea]
 != b[sizea]) return false;
  return true;
 }

 private static boolean invEqual(int inva[], int[] invb) {
  int i = inva.length;
  while(i-- != 0) if (inva[i] != invb[i]) return false;
  return true;
 }

 protected static void runTest(int n) {
  long ms;
  Exception mThrowsIllegal, tThrowsIllegal, mThrowsOutOfBounds, tThrowsOutOfBounds, mThrowsNoElement,
tThrowsNoElement;
  int rm = 0, rt = 0;
  KEY_TYPE[] refArray = new KEY_TYPE[n];

  for(int i = 0; i < n; i++) refArray[i] = genKey();

  HEAP_INDIRECT_PRIORITY_QUEUE m = new HEAP_INDIRECT_PRIORITY_QUEUE(refArray,
COMPARATORS.NATURAL_COMPARATOR);
  TestQueue t = new TestQueue(refArray, COMPARATORS.NATURAL_COMPARATOR,
COMPARATORS.NATURAL_COMPARATOR);

  /* We add pairs to t. */
  for(int i = 0; i < n / 2;  i++) {
   t.add(i);
   m.enqueue(i);
  }

  ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after creation (" + m + ",
" + t + ")");
  ensure(invEqual(m.inv, t.inSec), "Error (" +
 seed + "): m and t differ in inversion arrays after creation (" + java.util.Arrays.toString(m.inv) + ", " +
java.util.Arrays.toString(t.inSec) + ")");

  /* Now we add and remove random data in m and t, checking that the result is the same. */

  for(int i=0; i<2*n;  i++) {
   if (r.nextDouble() < 0.01) {
```

```
    t.clear();
    m.clear();
    for(int j = 0; j < n / 2;  j++) {
     t.add(j);
     m.enqueue(j);
    }
   }

   int T = r.nextInt(2 * n);

   mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

   try {
    m.enqueue(T);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { mThrowsIllegal = e; }

   try {
    t.add(T);
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { tThrowsIllegal = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): enqueue()
 divergence in IndexOutOfBoundsException for " + T + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
   ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): enqueue() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");

   ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after enqueue (" + m +
", " + t + ")");
   ensure(invEqual(m.inv, t.inSec), "Error (" + seed + "): m and t differ in inversion arrays after enqueue (" +
java.util.Arrays.toString(m.inv) + ", " + java.util.Arrays.toString(t.inSec) + ")");

   if (m.size() != 0) {
    ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after enqueue (" + m.first() + ", " +
t.top() + ")");
   }

   mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

   try {
    rm = m.dequeue();
   }
   catch (IndexOutOfBoundsException
```

```
 e) { mThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { mThrowsIllegal = e; }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = e; }

   try {
    rt = t.top();
    t.remove();
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { tThrowsIllegal = e; }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): dequeue()
divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");
   ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): dequeue() divergence in
IllegalArgumentException  (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
   ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): dequeue() divergence
in java.util.NoSuchElementException  (" + mThrowsNoElement
 + ", " + tThrowsNoElement + ")");
   if (mThrowsOutOfBounds == null) ensure(rt == rm , "Error (" + seed + "): divergence in dequeue() between t and m
(" + rt + ", " + rm + ")");

   ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after dequeue (" + m +
", " + t + ")");
   ensure(invEqual(m.inv, t.inSec), "Error (" + seed + "): m and t differ in inversion arrays after dequeue (" +
java.util.Arrays.toString(m.inv) + ", " + java.util.Arrays.toString(t.inSec) + ")");

   if (m.size() != 0) {
    ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after dequeue (" + m.first() + ", " +
t.top() + ")");
   }


   int pos = r.nextInt(n * 2);

   try {
    m.remove(pos);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { mThrowsIllegal = e; }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = e; }

   try {
    t.remove(pos);
   }
   catch
 (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { tThrowsIllegal = e; }
```

```
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): remove(int)
divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");
   ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): remove(int) divergence in
IllegalArgumentException  (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
   ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): remove(int)
divergence in java.util.NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
   if (mThrowsOutOfBounds == null) ensure(rt == rm , "Error (" + seed + "): divergence in remove(int) between t and
m (" + rt + ", " + rm + ")");

   ensure(heapEqual(m.heap, t.primary, m.size(), t.size()),
 "Error (" + seed + "): m and t differ after remove(int) (" + m + ", " + t + ")");
   ensure(invEqual(m.inv, t.inSec), "Error (" + seed + "): m and t differ in inversion arrays after remove(int) (" +
java.util.Arrays.toString(m.inv) + ", " + java.util.Arrays.toString(t.inSec) + ")");

   if (m.size() != 0) {
    ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after remove(int) (" + m.first() + ", "
+ t.top() + ")");
   }



   pos = r.nextInt(n * 2);

   try {
    m.changed(pos);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { mThrowsIllegal = e; }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = e; }

   try {
    t.change(pos);
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { tThrowsIllegal = e; }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = e; }

   ensure((mThrowsOutOfBounds ==
 null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): change(int) divergence in
IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");
   ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): change(int) divergence in
IllegalArgumentException  (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
   ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): change(int)
divergence in java.util.NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
   if (mThrowsOutOfBounds == null) ensure(rt == rm , "Error (" + seed + "): divergence in change(int) between t and
m (" + rt + ", " + rm + ")");
```

```
   ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after change(int) (" + m
+ ", " + t + ")");
   ensure(invEqual(m.inv, t.inSec), "Error (" + seed + "): m and t differ in inversion arrays after change(int) (" +
java.util.Arrays.toString(m.inv)
 + ", " + java.util.Arrays.toString(t.inSec) + ")");

   if (m.size() != 0) {
    ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after change(int) (" + m.first() + ", " +
t.top() + ")");
   }

   if (m.size() != 0) {

    refArray[m.first()] = genKey();

    m.changed();
    t.change();

    ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after change (" + m + ",
" + t + ")");
    ensure(invEqual(m.inv, t.inSec), "Error (" + seed + "): m and t differ in inversion arrays after change (" +
java.util.Arrays.toString(m.inv) + ", " + java.util.Arrays.toString(t.inSec) + ")");

    if (m.size() != 0) {
     ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after change (" + m.first() + ", " +
t.top() + ")");
    }
   }
  }


  /* Now we check that m actually holds the same data. */

  m.clear();
  ensure(m.isEmpty(), "Error (" + seed + "): m is not empty after clear()");

  System.out.println("Test
 OK");
 }



 public static void main(String args[]) {
  int n  = Integer.parseInt(args[1]);
  if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));


  try {
```

```
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
   else if ("test".equals(args[0])) runTest(n);
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed: " + seed);
  }
 }

#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/HeapIndirectPriorityQueue.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import it.unimi.dsi.fastutil.Hash;

/** A type-specific {@link Hash} interface.
*
* @see Hash
*/

public interface HASH {

 /** A type-specific hash strategy.
  *
  * @see it.unimi.dsi.fastutil.Hash.Strategy
```

```
 */

interface Strategy {

 /** Returns the hash code of the specified element with respect
to this hash strategy.
  *
  * @param e an element.
  * @return the hash code of the given element with respect to this hash strategy.
  */

 int hashCode(KEY_TYPE e);

 /** Returns true if the given elements are equal with respect to this hash strategy.
  *
  * @param a an element.
  * @param b another element.
  * @return true if the two specified elements are equal with respect to this hash strategy.
  */

 boolean equals(KEY_TYPE a, KEY_TYPE b);
}
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Hash.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;
```

```
/** A type-specific {@link Iterable} that further strengthens the specification of {@link Iterable#iterator()}.
*/
public interface KEY_BIDI_ITERABLE KEY_GENERIC extends KEY_ITERABLE KEY_GENERIC {

 /** Returns a type-specific {@link it.unimi.dsi.fastutil.BidirectionalIterator}.
  *
  * @return a type-specific
 bidirectional iterator.
  */
 @Override
 KEY_BIDI_ITERATOR KEY_GENERIC iterator();
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/BidirectionalIterable.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

#if KEY_CLASS_Object
import java.util.Comparator;
#endif

import it.unimi.dsi.fastutil.ints.IntArrays;

/** A class providing static methods and objects that do useful things with semi-indirect heaps.
*
* <p>A semi-indirect heap is based on a <em>reference array</em>. Elements of
*
```

```
 a semi-indirect heap are integers that index the reference array (note that
* in an <em>indirect</em> heap you can also map elements of the reference
* array to heap positions).
*/

public final class SEMI_INDIRECT_HEAPS {

 private SEMI_INDIRECT_HEAPS() {}

 /** Moves the given element down into the semi-indirect heap until it reaches the lowest possible position.
  *
  * @param refArray the reference array.
  * @param heap the semi-indirect heap (starting at 0).
  * @param size the number of elements in the heap.
  * @param i the index in the heap of the element to be moved down.
  * @param c a type-specific comparator, or {@code null} for the natural order.
  * @return the new position in the heap of the element of heap index {@code i}.
  */

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public static KEY_GENERIC int downHeap(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int size,
int i, final KEY_COMPARATOR KEY_GENERIC c) {
  assert i < size;

  final int e = heap[i];
  final
 KEY_GENERIC_TYPE E = refArray[e];
  int child;

  if (c == null)
   while ((child = (i << 1) + 1) < size) {
    int t = heap[child];
    final int right = child + 1;
    if (right < size && KEY_LESS(refArray[heap[right]], refArray[t])) t = heap[child = right];
    if (KEY_LESSEQ(E, refArray[t])) break;
    heap[i] = t;
    i = child;
   }
  else
   while ((child = (i << 1) + 1) < size) {
    int t = heap[child];
    final int right = child + 1;
    if (right < size && c.compare(refArray[heap[right]], refArray[t]) < 0) t = heap[child = right];
    if (c.compare(E, refArray[t]) <= 0) break;
    heap[i] = t;
    i = child;
   }
```

```
 heap[i] = e;

 return i;
}

/** Moves the given element up in the semi-indirect heap until it reaches the highest possible position.
 *
 * @param refArray the reference array.
 * @param heap the semi-indirect heap (starting at 0).
 * @param size the number of elements in the heap.
 * @param i the index in the heap of the element to be moved up.
 * @param
c a type-specific comparator, or {@code null} for the natural order.
 * @return the new position in the heap of the element of heap index {@code i}.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC int upHeap(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int size, int
i, final KEY_COMPARATOR KEY_GENERIC c) {
 assert i < size;

 final int e = heap[i];
 final KEY_GENERIC_TYPE E = refArray[e];

 if (c == null)
  while (i != 0) {
   final int parent = (i - 1) >>> 1;
   final int t = heap[parent];
   if (KEY_LESSEQ(refArray[t], E)) break;
   heap[i] = t;
   i = parent;
  }
 else
  while (i != 0) {
   final int parent = (i - 1) >>> 1;
   final int t = heap[parent];
   if (c.compare(refArray[t], E) <= 0) break;
   heap[i] = t;
   i = parent;
  }

 heap[i] = e;

 return i;
}

/** Creates a semi-indirect heap in the given array.
 *
```

```
 * @param refArray the reference array.
 * @param offset the first element of the reference array to be put
in the heap.
 * @param length the number of elements to be put in the heap.
 * @param heap the array where the heap is to be created.
 * @param c a type-specific comparator, or {@code null} for the natural order.
 */

public static KEY_GENERIC void makeHeap(final KEY_GENERIC_TYPE[] refArray, final int offset, final int
length, final int[] heap, final KEY_COMPARATOR KEY_GENERIC c) {
  ARRAYS.ensureOffsetLength(refArray, offset, length);
  if (heap.length < length) throw new IllegalArgumentException("The heap length (" + heap.length + ") is smaller
than the number of elements (" + length + ")");

  int i = length;
  while(i-- != 0) heap[i] = offset + i;

  i = length >>> 1;
  while(i-- != 0) downHeap(refArray, heap, length, i, c);
 }

/** Creates a semi-indirect heap, allocating its heap array.
 *
 * @param refArray the reference array.
 * @param offset the first element of the reference array to be put in the heap.
 * @param length the number of elements to be put in the heap.

* @param c a type-specific comparator, or {@code null} for the natural order.
 * @return the heap array.
 */

public static KEY_GENERIC int[] makeHeap(final KEY_GENERIC_TYPE[] refArray, final int offset, final int
length, final KEY_COMPARATOR KEY_GENERIC c) {
  final int[] heap = length <= 0 ? IntArrays.EMPTY_ARRAY : new int[length];
  makeHeap(refArray, offset, length, heap, c);
  return heap;
 }



/** Creates a semi-indirect heap from a given index array.
 *
 * @param refArray the reference array.
 * @param heap an array containing indices into {@code refArray}.
 * @param size the number of elements in the heap.
 * @param c a type-specific comparator, or {@code null} for the natural order.
 */
```

```
 public static KEY_GENERIC void makeHeap(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int
size, final KEY_COMPARATOR KEY_GENERIC c) {
  int i = size >>> 1;
  while(i-- != 0) downHeap(refArray, heap, size, i, c);
 }

 /** Retrieves the front of a heap in a given array.

 *
  * <p>The <em>front</em> of a semi-indirect heap is the set of indices whose associated elements in the reference
array
  * are equal to the element associated to the first index.
  *
  * <p>In several circumstances you need to know the front, and scanning linearly the entire heap is not
  * the best strategy. This method simulates (using a partial linear scan) a breadth-first visit that
  * terminates when all visited nodes are larger than the element associated
  * to the top index, which implies that no elements of the front can be found later.
  * In most cases this trick yields a significant improvement.
  *
  * @param refArray the reference array.
  * @param heap an array containing indices into {@code refArray}.
  * @param size the number of elements in the heap.
  * @param a an array large enough to hold the front (e.g., at least long as {@code refArray}).
  * @return the number of elements actually written (starting from the first position of {@code a}).
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public
 static KEY_GENERIC int front(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int size, final int[]
a) {
  final KEY_GENERIC_TYPE top = refArray[heap[0]];
  int j = 0, // The current position in a
   l = 0, // The first position to visit in the next level (inclusive)
   r = 1, // The last position to visit in the next level (exclusive)
   f = 0; // The first position (in the heap array) of the next level
  for(int i = 0; i < r; i++) {
   if (i == f) { // New level
    if (l >= r) break; // If we are crossing the two bounds, we're over
    f = (f << 1) + 1; // Update the first position of the next level...
    i = l; // ...and jump directly to position l
    l = -1; // Invalidate l
   }
   if (KEY_CMP_EQ(top, refArray[heap[i]])) {
    a[j++] = heap[i];
    if (l == -1) l = i * 2 + 1; // If this is the first time in this level, set l
    r = Math.min(size, i * 2 + 3); // Update r, but do not go beyond size
   }
  }
```

```
 return j;
}

/** Retrieves
the front of a heap in a given array using a given comparator.
 *
 * <p>The <em>front</em> of a semi-indirect heap is the set of indices whose associated elements in the reference
array
 * are equal to the element associated to the first index.
 *
 * <p>In several circumstances you need to know the front, and scanning linearly the entire heap is not
 * the best strategy. This method simulates (using a partial linear scan) a breadth-first visit that
 * terminates when all visited nodes are larger than the element associated
 * to the top index, which implies that no elements of the front can be found later.
 * In most cases this trick yields a significant improvement.
 *
 * @param refArray the reference array.
 * @param heap an array containing indices into {@code refArray}.
 * @param size the number of elements in the heap.
 * @param a an array large enough to hold the front (e.g., at least long as {@code refArray}).
 * @param c a type-specific comparator.
 * @return
the number of elements actually written (starting from the first position of {@code a}).
 */
public static KEY_GENERIC int front(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int size, final
int[] a, final KEY_COMPARATOR KEY_GENERIC c) {
 final KEY_GENERIC_TYPE top = refArray[heap[0]];
 int j = 0, // The current position in a
  l = 0, // The first position to visit in the next level (inclusive)
  r = 1, // The last position to visit in the next level (exclusive)
  f = 0; // The first position (in the heap array) of the next level
 for(int i = 0; i < r; i++) {
  if (i == f) { // New level
   if (l >= r) break; // If we are crossing the two bounds, we're over
   f = (f << 1) + 1; // Update the first position of the next level...
   i = l; // ...and jump directly to position l
   l = -1; // Invalidate l
  }
  if (c.compare(top, refArray[heap[i]]) == 0) {
   a[j++] = heap[i];
   if (l == -1) l = i * 2 + 1; // If this is the first time in this level, set l
   r
= Math.min(size, i * 2 + 3); // Update r, but do not go beyond size
  }
 }

 return j;
}
```

```
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/SemiIndirectHeaps.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import it.unimi.dsi.fastutil.Stack;

/** A type-specific {@link Stack}; provides some additional methods that use polymorphism to avoid (un)boxing.
*/

public interface STACK KEY_GENERIC extends Stack<KEY_GENERIC_CLASS> {

 /** Pushes the given object on the stack.
  * @param k the object to push on
 the stack.
  * @see Stack#push(Object)
  */
 void push(KEY_TYPE k);

 /** Pops the top off the stack.
  *
  * @return the top of the stack.
  * @see Stack#pop()
  */
 KEY_TYPE POP();

 /** Peeks at the top of the stack (optional operation).
```

```
 * @return the top of the stack.
 * @see Stack#top()
 */
KEY_TYPE TOP();

/** Peeks at an element on the stack (optional operation).
 * @param i an index from the stop of the stack (0 represents the top).
 * @return the <code>i</code>-th element on the stack.
 * @see Stack#peek(int)
 */
KEY_TYPE PEEK(int i);

/** {@inheritDoc}
 * <p>This default implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default void push(KEY_GENERIC_CLASS o) {
 push(o.KEY_VALUE());
}

/** {@inheritDoc}
 * <p>This default implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific
method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS pop() {
 return KEY_CLASS.valueOf(POP());
}

/** {@inheritDoc}
 * <p>This default implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS top() {
 return KEY_CLASS.valueOf(TOP());
}

/** {@inheritDoc}
 * <p>This default implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS peek(final int i) {
 return KEY_CLASS.valueOf(PEEK(i));
}
```

}

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Stack.drv
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2003-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/PriorityQueues.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/IndirectPriorityQueues.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package PACKAGE;

```
import java.util.Collection;
import java.util.Iterator;
import java.util.RandomAccess;
import java.util.NoSuchElementException;
import it.unimi.dsi.fastutil.BigArrays;
import static it.unimi.dsi.fastutil.BigArrays.length;

#if KEYS_PRIMITIVE

/** A type-specific big list based on a big array; provides
 some additional methods that use polymorphism to avoid (un)boxing.
*
* <p>This class implements a lightweight, fast, open, optimized,
* reuse-oriented version of big-array-based big lists. Instances of this class
* represent a big list with a big array that is enlarged as needed when new entries
* are created (by doubling the current length), but is
* <em>never</em> made smaller (even on a {@link #clear()}). A family of
* {@linkplain #trim() trimming methods} lets you control the size of the
* backing big array; this is particularly useful if you reuse instances of this class.
* Range checks are equivalent to those of {@link java.util}'s classes, but
* they are delayed as much as possible. The backing big array is exposed by the
* {@link #elements()} method.
*
* <p>This class implements the bulk methods {@code removeElements()},
* {@code addElements()} and {@code getElements()} using
* high-performance system calls (e.g., {@link
* System#arraycopy(Object,int,Object,int,int)
 System.arraycopy()} instead of
* expensive loops.
*
* @see java.util.ArrayList
*/

public class BIG_ARRAY_BIG_LIST KEY_GENERIC extends ABSTRACT_BIG_LIST KEY_GENERIC
implements RandomAccess, Cloneable, java.io.Serializable {
 private static final long serialVersionUID = -7046029254386353130L;


#else

/** A type-specific big-array-based big list; provides some additional methods that use polymorphism to avoid
(un)boxing.
*
* <p>This class implements a lightweight, fast, open, optimized,
* reuse-oriented version of big-array-based big lists. Instances of this class
* represent a big list with a big array that is enlarged as needed when new entries
* are created (by doubling the current length), but is
```

```
* <em>never</em> made smaller (even on a {@link #clear()}). A family of
* {@linkplain #trim() trimming methods} lets you control the size of the
* backing big array; this is particularly useful if you reuse instances of this class.
* Range checks are equivalent to those
 of {@link java.util}'s classes, but
* they are delayed as much as possible.
*
* <p>The backing big array is exposed by the {@link #elements()} method. If an instance
* of this class was created {@linkplain #wrap(Object[][],long) by wrapping},
* backing-array reallocations will be performed using reflection, so that
* {@link #elements()} can return a big array of the same type of the original big array; the comments
* about efficiency made in {@link it.unimi.dsi.fastutil.objects.ObjectArrays} apply here.
*
* <p>This class implements the bulk methods {@code removeElements()},
* {@code addElements()} and {@code getElements()} using
* high-performance system calls (e.g., {@link
* System#arraycopy(Object,int,Object,int,int) System.arraycopy()} instead of
* expensive loops.
*
* @see java.util.ArrayList
*/

public class BIG_ARRAY_BIG_LIST KEY_GENERIC extends ABSTRACT_BIG_LIST KEY_GENERIC
implements RandomAccess, Cloneable, java.io.Serializable {
 private static final long serialVersionUID
 = -7046029254386353131L;


#endif

 /** The initial default capacity of a big-array big list. */
 public static final int DEFAULT_INITIAL_CAPACITY = 10;

#if ! KEYS_PRIMITIVE
 /** Whether the backing big array was passed to {@code wrap()}. In
  * this case, we must reallocate with the same type of big array. */
 protected final boolean wrapped;
#endif

 /** The backing big array. */
 protected transient KEY_GENERIC_TYPE a[][];

 /** The current actual size of the big list (never greater than the backing-array length). */
 protected long size;

 /** Creates a new big-array big list using a given array.
  *
  * <p>This constructor is only meant to be used by the wrapping methods.
```

```
  *
  * @param a the big array that will be used to back this big-array big list.
  */

 protected BIG_ARRAY_BIG_LIST(final KEY_GENERIC_TYPE a[][], @SuppressWarnings("unused") boolean
dummy) {
  this.a = a;
#if ! KEYS_PRIMITIVE
  this.wrapped = true;
#endif
 }

 /** Creates a new big-array big list
 with given capacity.
  *
  * @param capacity the initial capacity of the array list (may be 0).
  */

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public BIG_ARRAY_BIG_LIST(final long capacity) {
  if (capacity < 0) throw new IllegalArgumentException("Initial capacity (" + capacity + ") is negative");
  if (capacity == 0) a = KEY_GENERIC_BIG_ARRAY_CAST BIG_ARRAYS.EMPTY_BIG_ARRAY;
  else a = KEY_GENERIC_BIG_ARRAY_CAST BIG_ARRAYS.newBigArray(capacity);
#if ! KEYS_PRIMITIVE
  wrapped = false;
#endif
 }

 /** Creates a new big-array big list with {@link #DEFAULT_INITIAL_CAPACITY} capacity. */

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public BIG_ARRAY_BIG_LIST() {
  a = KEY_GENERIC_BIG_ARRAY_CAST BIG_ARRAYS.DEFAULT_EMPTY_BIG_ARRAY; // We delay
allocation
#if ! KEYS_PRIMITIVE
  wrapped = false;
#endif
 }

 /** Creates a new big-array big list and fills it with a given type-specific collection.
  *
  * @param c a type-specific collection that will be used to fill the array list.
  */

 public BIG_ARRAY_BIG_LIST(final
 COLLECTION KEY_EXTENDS_GENERIC c) {
  this(c.size());
  for(KEY_ITERATOR KEY_EXTENDS_GENERIC i = c.iterator(); i.hasNext();) add(i.NEXT_KEY());
```

```
}

/** Creates a new big-array big list and fills it with a given type-specific list.
 *
 * @param l a type-specific list that will be used to fill the array list.
 */

public BIG_ARRAY_BIG_LIST(final BIG_LIST KEY_EXTENDS_GENERIC l) {
 this(l.size64());
 l.getElements(0, a, 0, size = l.size64());
}

/** Creates a new big-array big list and fills it with the elements of a given big array.
 *
 * <p>Note that this constructor makes it easy to build big lists from literal arrays
 * declared as <code><var>type</var>[][] {{ <var>init_values</var> }}</code>.
 * The only constraint is that the number of initialisation values is
 * below {@link it.unimi.dsi.fastutil.BigArrays#SEGMENT_SIZE}.
 *
 * @param a a big array whose elements will be used to fill the array list.
 */

public BIG_ARRAY_BIG_LIST(final KEY_GENERIC_TYPE a[][])
{
 this(a, 0, length(a));
}

/** Creates a new big-array big list and fills it with the elements of a given big array.
 *
 * <p>Note that this constructor makes it easy to build big lists from literal arrays
 * declared as <code><var>type</var>[][] {{ <var>init_values</var> }}</code>.
 * The only constraint is that the number of initialisation values is
 * below {@link it.unimi.dsi.fastutil.BigArrays#SEGMENT_SIZE}.
 *
 * @param a a big array whose elements will be used to fill the array list.
 * @param offset the first element to use.
 * @param length the number of elements to use.
 */

public BIG_ARRAY_BIG_LIST(final KEY_GENERIC_TYPE a[][], final long offset, final long length) {
 this(length);
 BigArrays.copy(a, offset, this.a, 0, length);
 size = length;
}

/** Creates a new big-array big list and fills it with the elements returned by an iterator..
 *
 * @param i an iterator whose returned elements will fill the array list.
```

```
 */

 public BIG_ARRAY_BIG_LIST(final
 Iterator<? extends KEY_GENERIC_CLASS> i) {
  this();
  while(i.hasNext()) this.add(KEY_CLASS2TYPE(i.next()));
 }

 /** Creates a new big-array big list and fills it with the elements returned by a type-specific iterator..
  *
  * @param i a type-specific iterator whose returned elements will fill the array list.
  */

 public BIG_ARRAY_BIG_LIST(final KEY_ITERATOR KEY_EXTENDS_GENERIC i) {
  this();
  while(i.hasNext()) this.add(i.NEXT_KEY());
 }

#if KEYS_PRIMITIVE
 /** Returns the backing big array of this big list.
  *
  * @return the backing big array.
  */

 public KEY_GENERIC_TYPE[][] elements() {
  return a;
 }
#else
 /** Returns the backing big array of this big list.
  *
  * <p>If this big-array big list was created by wrapping a given big array, it is guaranteed
  * that the type of the returned big array will be the same. Otherwise, the returned
  * big array will be an big array of objects.
  *
  * @return the backing big array.
  */

 public KEY_GENERIC_TYPE[][] elements()
 {
  return a;
 }
#endif

 /** Wraps a given big array into a big-array list of given size.
  *
  * @param a a big array to wrap.
  * @param length the length of the resulting big-array list.
  * @return a new big-array list of the given size, wrapping the given big array.
```

```
  */

 public static KEY_GENERIC BIG_ARRAY_BIG_LIST KEY_GENERIC wrap(final KEY_GENERIC_TYPE a[][],
final long length) {
  if (length > length(a)) throw new IllegalArgumentException("The specified length (" + length + ") is greater than the
array size (" + length(a) + ")");
  final BIG_ARRAY_BIG_LIST KEY_GENERIC l = new BIG_ARRAY_BIG_LIST
KEY_GENERIC_DIAMOND(a, false);
  l.size = length;
  return l;
 }

 /** Wraps a given big array into a big-array big list.
  *
  * @param a a big array to wrap.
  * @return a new big-array big list wrapping the given array.
  */

 public static KEY_GENERIC BIG_ARRAY_BIG_LIST KEY_GENERIC wrap(final KEY_GENERIC_TYPE
a[][]) {
  return wrap(a, length(a));
 }


 /** Ensures that this big-array
 big list can contain the given number of entries without resizing.
  *
  * @param capacity the new minimum capacity for this big-array big list.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public void ensureCapacity(final long capacity) {
  if (capacity <= length(a) || a == BIG_ARRAYS.DEFAULT_EMPTY_BIG_ARRAY) return;
#if KEYS_PRIMITIVE
  a = BigArrays.forceCapacity(a, capacity, size);
#else
  if (wrapped) a = BigArrays.forceCapacity(a, capacity, size);
  else {
   if (capacity > length(a)) {
    final Object t[][] = BIG_ARRAYS.newBigArray(capacity);
    BigArrays.copy(a, 0, t, 0, size);
    a = (KEY_GENERIC_TYPE[][])t;
   }
  }
#endif
  assert size <= length(a);
 }

 /** Grows this big-array big list, ensuring that it can contain the given number of entries without resizing,
```

```
 * and in case increasing current capacity at least by a factor of 50%.
 *
 * @param capacity the new minimum capacity for this big-array big list.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
private void grow(long
capacity) {
 final long oldLength = length(a);
 if (capacity <= oldLength) return;
 if (a != BIG_ARRAYS.DEFAULT_EMPTY_BIG_ARRAY) capacity = Math.max(oldLength + (oldLength >> 1),
capacity);
 else if (capacity < DEFAULT_INITIAL_CAPACITY) capacity = DEFAULT_INITIAL_CAPACITY;
#if KEYS_PRIMITIVE
 a = BigArrays.forceCapacity(a, capacity, size);
#else
 if (wrapped) a =  BigArrays.forceCapacity(a, capacity, size);
 else {
  final Object t[][] = BIG_ARRAYS.newBigArray(capacity);
  BigArrays.copy(a, 0, t, 0, size);
  a = (KEY_GENERIC_TYPE[][])t;
 }
#endif
 assert size <= length(a);
}

@Override
public void add(final long index, final KEY_GENERIC_TYPE k) {
 ensureIndex(index);
 grow(size + 1);
 if (index != size) BigArrays.copy(a, index, a, index + 1, size - index);
 BigArrays.set(a, index, k);
 size++;
 assert size <= length(a);
}

@Override
public boolean add(final KEY_GENERIC_TYPE k) {
 grow(size + 1);
 BigArrays.set(a, size++, k);
 assert size <= length(a);
 return
true;
}

@Override
public KEY_GENERIC_TYPE GET_KEY(final long index) {
 if (index >= size) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list size
(" + size + ")");
```

```
 return BigArrays.get(a, index);
 }

 @Override
 public long indexOf(final KEY_TYPE k) {
  for(long i = 0; i < size; i++) if (KEY_EQUALS(k, BigArrays.get(a, i))) return i;
  return -1;
 }

 @Override
 public long lastIndexOf(final KEY_TYPE k) {
  for(long i = size; i-- != 0;) if (KEY_EQUALS(k, BigArrays.get(a, i))) return i;
  return -1;
 }

 @Override
 public KEY_GENERIC_TYPE REMOVE_KEY(final long index) {
  if (index >= size) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list size
(" + size + ")");
  final KEY_GENERIC_TYPE old = BigArrays.get(a, index);
  size--;
  if (index != size) BigArrays.copy(a, index + 1, a, index, size - index);
#if KEYS_REFERENCE
  BigArrays.set(a, size, null);
#endif
  assert size <= length(a);
  return old;
 }

 @Override
 public
 boolean REMOVE(final KEY_TYPE k) {
  final long index = indexOf(k);
  if (index == -1) return false;
  REMOVE_KEY(index);
  assert size <= length(a);
  return true;
 }

 @Override
 public KEY_GENERIC_TYPE set(final long index, final KEY_GENERIC_TYPE k) {
  if (index >= size) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list size
(" + size + ")");
  KEY_GENERIC_TYPE old = BigArrays.get(a, index);
  BigArrays.set(a, index, k);
  return old;
 }
```

```
#if KEYS_PRIMITIVE
 @Override
 public boolean removeAll(final COLLECTION c) {
  KEY_GENERIC_TYPE[] s = null, d = null;
  int ss = -1, sd = BigArrays.SEGMENT_SIZE, ds = -1, dd = BigArrays.SEGMENT_SIZE;
  for (long i = 0; i < size; i++) {
   if (sd == BigArrays.SEGMENT_SIZE) {
    sd = 0;
    s = a[++ss];
   }
   if (!c.contains(s[sd])) {
    if (dd == BigArrays.SEGMENT_SIZE) {
     d = a[++ds];
     dd = 0;
    }
    d[dd++] = s[sd];
   }
   sd++;
  }
  final long j = BigArrays.index(ds,
 dd);
#if KEYS_REFERENCE
  BigArrays.fill(a, j, size, null);
#endif
  final boolean modified = size != j;
  size = j;
  return modified;
 }

#endif

 @Override
 public boolean removeAll(final Collection<?> c) {
  KEY_GENERIC_TYPE[] s = null, d = null;
  int ss = -1, sd = BigArrays.SEGMENT_SIZE, ds = -1, dd = BigArrays.SEGMENT_SIZE;
  for (long i = 0; i < size; i++) {
   if (sd == BigArrays.SEGMENT_SIZE) {
    sd = 0;
    s = a[++ss];
   }
   if (!c.contains(KEY2OBJ(s[sd]))) {
    if (dd == BigArrays.SEGMENT_SIZE) {
     d = a[++ds];
     dd = 0;
    }
    d[dd++] = s[sd];
   }
   sd++;
```

```
  }
  final long j = BigArrays.index(ds, dd);
#if KEYS_REFERENCE
  BigArrays.fill(a, j, size, null);
#endif
  final boolean modified = size != j;
  size = j;
  return modified;
 }

 @Override
 public void clear() {
#if KEYS_REFERENCE
  BigArrays.fill(a, 0, size, null);
#endif
  size = 0;
  assert size <= length(a);
 }

 @Override
 public long size64() {
  return size;
 }

 @Override
 public void size(final
 long size) {
  if (size > length(a)) a = BigArrays.forceCapacity(a, size, this.size);
  if (size > this.size) BigArrays.fill(a, this.size, size, KEY_NULL);
#if KEYS_REFERENCE
  else BigArrays.fill(a, size, this.size, KEY_NULL);
#endif
  this.size = size;
 }

 @Override
 public boolean isEmpty() {
  return size == 0;
 }

 /** Trims this big-array big list so that the capacity is equal to the size.
  *
  * @see java.util.ArrayList#trimToSize()
  */
 public void trim() {
  trim(0);
 }
```

```
/** Trims the backing big array if it is too large.
 *
 * If the current big array length is smaller than or equal to
 * {@code n}, this method does nothing. Otherwise, it trims the
 * big-array length to the maximum between {@code n} and {@link #size64()}.
 *
 * <p>This method is useful when reusing big lists.  {@linkplain #clear() Clearing a
 * big list} leaves the big-array length untouched. If you are reusing a big list
 * many times, you can call this method with a typical
 * size to
avoid keeping around a very large big array just
 * because of a few large transient big lists.
 *
 * @param n the threshold for the trimming.
 */

public void trim(final long n) {
 final long arrayLength = length(a);
 if (n >= arrayLength || size == arrayLength) return;
 a = BigArrays.trim(a, Math.max(n, size));
 assert size <= length(a);
}


  /** Copies element of this type-specific list into the given big array using optimized system calls.
 *
 * @param from the start index (inclusive).
 * @param a the destination big array.
 * @param offset the offset into the destination array where to store the first element copied.
 * @param length the number of elements to be copied.
 */

@Override
public void getElements(final long from, final KEY_TYPE[][] a, final long offset, final long length) {
 BigArrays.copy(this.a, from, a, offset, length);
}

/** Removes elements of this type-specific list using optimized system calls.
 *
 * @param from the start index (inclusive).

* @param to the end index (exclusive).
 */
@Override
public void removeElements(final long from, final long to) {
 BigArrays.ensureFromTo(size, from, to);
 BigArrays.copy(a, to, a, from, size - to);
 size -= (to - from);
```

```
#if KEYS_REFERENCE
  BigArrays.fill(a, size, size + to - from, null);
#endif
 }


 /** Adds elements to this type-specific list using optimized system calls.
  *
  * @param index the index at which to add elements.
  * @param a the big array containing the elements.
  * @param offset the offset of the first element to add.
  * @param length the number of elements to add.
  */
 @Override
 public void addElements(final long index, final KEY_GENERIC_TYPE a[][], final long offset, final long length) {
  ensureIndex(index);
  BigArrays.ensureOffsetLength(a, offset, length);
  grow(size + length);
  BigArrays.copy(this.a, index, this.a, index + length, size - index);
  BigArrays.copy(a, offset, this.a, index, length);
  size += length;
 }

 @Override
 public KEY_BIG_LIST_ITERATOR
 KEY_GENERIC listIterator(final long index) {
  ensureIndex(index);

  return new KEY_BIG_LIST_ITERATOR KEY_GENERIC() {
    long pos = index, last = -1;

    @Override
    public boolean hasNext() { return pos < size; }
    @Override
    public boolean hasPrevious() { return pos > 0; }
    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() { if (! hasNext()) throw new NoSuchElementException(); return
BigArrays.get(a, last = pos++); }
    @Override
    public KEY_GENERIC_TYPE PREV_KEY() { if (! hasPrevious()) throw new NoSuchElementException(); return
BigArrays.get(a, last = --pos); }
    @Override
    public long nextIndex() { return pos; }
    @Override
    public long previousIndex() { return pos - 1; }
    @Override
    public void add(KEY_GENERIC_TYPE k) {
     BIG_ARRAY_BIG_LIST.this.add(pos++, k);
```

```
    last = -1;
   }
   @Override
   public void set(KEY_GENERIC_TYPE k) {
    if (last == -1) throw new IllegalStateException();
    BIG_ARRAY_BIG_LIST.this.set(last,
 k);
   }
   @Override
   public void remove() {
    if (last == -1) throw new IllegalStateException();
    BIG_ARRAY_BIG_LIST.this.REMOVE_KEY(last);
    /* If the last operation was a next(), we are removing an element *before* us, and we must decrease pos
correspondingly. */
    if (last < pos) pos--;
    last = -1;
   }
  };
 }

 @Override
 public BIG_ARRAY_BIG_LIST KEY_GENERIC clone() {
  BIG_ARRAY_BIG_LIST KEY_GENERIC c = new BIG_ARRAY_BIG_LIST
KEY_GENERIC_DIAMOND(size);
  BigArrays.copy(a, 0, c.a, 0, size);
  c.size = size;
  return c;
 }

#if KEY_CLASS_Object
 private boolean valEquals(final K a, final K b) {
  return a == null ? b == null : a.equals(b);
 }
#endif

 /** Compares this type-specific big-array list to another one.
  *
  * <p>This method exists only for sake of efficiency. The implementation
  * inherited from the abstract implementation would already work.
  *
  * @param l a type-specific big-array list.
  * @return true if the argument contains the same
 elements of this type-specific big-array list.
  */
 public boolean equals(final BIG_ARRAY_BIG_LIST KEY_GENERIC l) {
  if (l == this) return true;
  long s = size64();
  if (s != l.size64()) return false;
```

```
  final KEY_GENERIC_TYPE[][] a1 = a;
  final KEY_GENERIC_TYPE[][] a2 = l.a;

#if KEY_CLASS_Object
  while(s-- !=  0) if (! valEquals(BigArrays.get(a1, s), BigArrays.get(a2, s))) return false;
#else
  while(s-- !=  0) if (BigArrays.get(a1, s) != BigArrays.get(a2, s)) return false;
#endif
  return true;
 }


#if ! KEY_CLASS_Reference

 /** Compares this big list to another big list.
  *
  * <p>This method exists only for sake of efficiency. The implementation
  * inherited from the abstract implementation would already work.
  *
  * @param l a big list.
  * @return a negative integer,
  * zero, or a positive integer as this big list is lexicographically less than, equal
  * to, or greater than the argument.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public int compareTo(final BIG_ARRAY_BIG_LIST
 KEY_EXTENDS_GENERIC l) {
  final long s1 = size64(), s2 = l.size64();
  final KEY_GENERIC_TYPE a1[][] = a, a2[][] = l.a;
  KEY_GENERIC_TYPE e1, e2;
  int r, i;

  for(i = 0; i < s1 && i < s2; i++) {
   e1 = BigArrays.get(a1, i);
   e2 = BigArrays.get(a2, i);
   if ((r = KEY_CMP(e1, e2)) != 0) return r;
  }

  return i < s2 ? -1 : (i < s1 ? 1 : 0);
 }
#endif


 private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
  s.defaultWriteObject();
  for(int i = 0; i < size; i++) s.WRITE_KEY(BigArrays.get(a, i));
 }
```

```
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
  s.defaultReadObject();
  a = KEY_GENERIC_BIG_ARRAY_CAST BIG_ARRAYS.newBigArray(size);
  for(int i = 0; i < size; i++) BigArrays.set(a, i, KEY_GENERIC_CAST s.READ_KEY());
 }


#ifdef TEST

 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);

 private
 static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif KEY_CLASS_Object
  return Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }

 private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
 private static java.text.FieldPosition p = new java.text.FieldPosition(0);

 private static String format(double d) {
  StringBuffer s = new StringBuffer();
  return format.format(d, s, p).toString();
 }

 private static void speedTest(int n, boolean comp) {
  System.out.println("There are presently no speed tests for this class.");
 }


 private static void fatal(String msg) {
  System.out.println(msg);
  System.exit(1);
 }

 private static void ensure(boolean cond, String msg) {
  if (cond) return;
  fatal(msg);
 }
```

```
 private static Object[] k, v, nk;
 private static KEY_TYPE kt[];
 private static
 KEY_TYPE nkt[];
 private static BIG_ARRAY_BIG_LIST topList;

 protected static void testLists(BIG_LIST m, BIG_LIST t, int n, int level) {
  long ms;
  Exception mThrowsIllegal, tThrowsIllegal, mThrowsOutOfBounds, tThrowsOutOfBounds;
  Object rt = null;
  KEY_TYPE rm = KEY_NULL;

  if (level > 4) return;


  /* Now we check that both sets agree on random keys. For m we use the polymorphic method. */

  for(int i = 0; i < n; i++) {
   int p = r.nextInt() % (n * 2);

   KEY_TYPE T = genKey();
   mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

   try {
    m.set(p, T);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   try {
    t.set(p, KEY2OBJ(T));
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
set() divergence at start in IndexOutOfBoundsException for index " + p + "  (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
   if
 (mThrowsOutOfBounds == null) ensure(t.get(p).equals(KEY2OBJ(m.GET_KEY(p))), "Error (" + level + ", " +
seed + "): m and t differ after set() on position " + p + " (" + m.GET_KEY(p) + ", " + t.get(p) + ")");

   p = r.nextInt() % (n * 2);
   mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

   try {
    m.GET_KEY(p);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   try {
```

```
    t.get(p);
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
get() divergence at start in IndexOutOfBoundsException for index " + p + "  (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
   if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(KEY2OBJ(m.GET_KEY(p))), "Error (" + level + ", " +
seed + "): m and t differ aftre get() on position " + p + " (" + m.GET_KEY(p) + ", " + t.get(p) + ")");

  }

  /* Now we check that both sets agree on
 random keys. For m we use the standard method. */

  for(int i = 0; i < n; i++) {
   int p = r.nextInt() % (n * 2);

   mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

   try {
    m.get(p);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   try {
    t.get(p);
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
get() divergence at start in IndexOutOfBoundsException for index " + p + "  (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
   if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(m.get(p)), "Error (" + level + ", " + seed + "): m and t
differ at start on position " + p + " (" + m.get(p) + ", " + t.get(p) + ")");

  }

  /* Now we check that m and t are equal. */
  if (!m.equals(t) || ! t.equals(m)) System.err.println("m: " + m + " t: " + t);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t)
 at start");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");



  /* Now we check that m actually holds that data. */
  for(Iterator i=t.iterator(); i.hasNext();) {
```

```
  ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */
  for(Iterator i=m.listIterator(); i.hasNext();) {
  ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on m)");
  }

  /* Now we check that inquiries about random data give the same answer in m and t. For
     m we use the polymorphic method. */

  for(int i=0; i<n;  i++) {
  KEY_TYPE T = genKey();
  ensure(m.contains(T) == t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in content between t and m (polymorphic method)");
  }

  /* Again, we check that inquiries about random data
 give the same answer in m and t, but
     for m we use the standard method. */

  for(int i=0; i<n;  i++) {
  KEY_TYPE T = genKey();
  ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in content between t and m (polymorphic method)");
  }

  /* Now we add and remove random data in m and t, checking that the result is the same. */

  for(int i=0; i<2*n;  i++) {
  KEY_TYPE T = genKey();

  try {
   m.add(T);
  }
  catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

  try {
   t.add(KEY2OBJ(T));
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

  T = genKey();
  int p = r.nextInt() % (2 * n + 1);

  mThrowsOutOfBounds = tThrowsOutOfBounds  = null;
```

```
  try {
   m.add(p, T);
  }
  catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

  try {
   t.add(p, KEY2OBJ(T));
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }


  ensure((mThrowsOutOfBounds
 == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "): add() divergence in
IndexOutOfBoundsException for index " + p + " for " + T + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");

  p = r.nextInt() % (2 * n + 1);

  mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

  try {
   rm = m.REMOVE_KEY(p);
  }
  catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

  try {
   rt = t.remove(p);
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }


  ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
remove() divergence in IndexOutOfBoundsException for index " + p + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
  if (mThrowsOutOfBounds == null) ensure(rt.equals(KEY2OBJ(rm)), "Error (" + level + ", " + seed + "): divergence
in remove() between t and m (" + rt + ", " + rm + ")");
  }

 ensure(m.equals(t), "Error (" + level + ", " + seed + "): !
 m.equals(t) after add/remove");
 ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after add/remove");

 /* Now we add random data in m and t using addAll on a collection, checking that the result is the same. */

 for(int i=0; i<n;  i++) {
  int p = r.nextInt() % (2 * n + 1);
  java.util.Collection m1 = new java.util.ArrayList();
```

```
  int s = r.nextInt(n / 2 + 1);
  for(int j = 0; j < s; j++) m1.add(KEY2OBJ(genKey()));

  mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

  try {
   m.addAll(p, m1);
  }
  catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

  try {
   t.addAll(p, m1);
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

  ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
addAll() divergence in IndexOutOfBoundsException for index " + p + " for " + m1 + " (" + mThrowsOutOfBounds
+ ", " + tThrowsOutOfBounds + ")");

  ensure(m.equals(t), "Error (" +
 level + ", " + seed + m + t + "): ! m.equals(t) after addAll");
  ensure(t.equals(m), "Error (" + level + ", " + seed + m + t + "): ! t.equals(m) after addAll");
  }

 if (m.size64() > n) {
  m.size(n);
  while(t.size64() != n) t.remove(t.size64() -1);
  }

 /* Now we add random data in m and t using addAll on a type-specific collection, checking that the result is the
same. */

 for(int i=0; i<n;  i++) {
  int p = r.nextInt() % (2 * n + 1);
  COLLECTION m1 = new BIG_ARRAY_BIG_LIST();
  java.util.Collection t1 = new java.util.ArrayList();
  int s = r.nextInt(n / 2 + 1);
  for(int j = 0; j < s; j++) {
   KEY_TYPE x = genKey();
   m1.add(x);
   t1.add(KEY2OBJ(x));
  }

  mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

  try {
   m.addAll(p, m1);
  }
```

```
  catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

  try {
   t.addAll(p, t1);
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

  ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds
 == null), "Error (" + level + ", " + seed + "): polymorphic addAll() divergence in IndexOutOfBoundsException for
index " + p + " for " + m1 + " (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");

  ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after polymorphic addAll");
  ensure(t.equals(m), "Error (" + level + ", " + seed + m + t + "): ! t.equals(m) after polymorphic addAll");
  }

 if (m.size64() > n) {
  m.size(n);
  while(t.size64() != n) t.remove(t.size64() -1);
  }

 /* Now we add random data in m and t using addAll on a list, checking that the result is the same. */

 for(int i=0; i<n;  i++) {
  int p = r.nextInt() % (2 * n + 1);
  BIG_LIST m1 = new BIG_ARRAY_BIG_LIST();
  java.util.Collection t1 = new java.util.ArrayList();
  int s = r.nextInt(n / 2 + 1);
  for(int j = 0; j < s; j++) {
   KEY_TYPE x = genKey();
   m1.add(x);
   t1.add(KEY2OBJ(x));
  }

  mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

  try
 {
   m.addAll(p, m1);
  }
  catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

  try {
   t.addAll(p, t1);
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

  ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "): list
addAll() divergence in IndexOutOfBoundsException for index " + p + " for " + m1 + " (" + mThrowsOutOfBounds
```

```
+ ", " + tThrowsOutOfBounds + ")");

   ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after list addAll");
   ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after list addAll");
  }

  /* Now we check that both sets agree on random keys. For m we use the standard method. */

  for(int i = 0; i < n; i++) {
   int p = r.nextInt() % (n * 2);

   mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

   try {
    m.get(p);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   try {
    t.get(p);
   }
   catch
 (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
get() divergence in IndexOutOfBoundsException for index " + p + "  (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
   if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(m.get(p)), "Error (" + level + ", " + seed + "): m and t
differ on position " + p + " (" + m.get(p) + ", " + t.get(p) +")");

  }

  /* Now we inquiry about the content with indexOf()/lastIndexOf(). */

  for(int i=0; i<10*n;  i++) {
   KEY_TYPE T = genKey();
   ensure(m.indexOf(KEY2OBJ(T)) == t.indexOf(KEY2OBJ(T)),
     "Error (" + level + ", " + seed + "): indexOf() divergence for " + T + "  (" + m.indexOf(KEY2OBJ(T)) + ", " +
t.indexOf(KEY2OBJ(T)) + ")");
   ensure(m.lastIndexOf(KEY2OBJ(T)) == t.lastIndexOf(KEY2OBJ(T)),
     "Error (" + level + ", " + seed + "): lastIndexOf() divergence for " + T + "  (" + m.lastIndexOf(KEY2OBJ(T))
 + ", " + t.lastIndexOf(KEY2OBJ(T)) + ")");
   ensure(m.indexOf(T) == t.indexOf(KEY2OBJ(T)),
     "Error (" + level + ", " + seed + "): polymorphic indexOf() divergence for " + T + "  (" + m.indexOf(T) + ", " +
t.indexOf(KEY2OBJ(T)) + ")");
   ensure(m.lastIndexOf(T) == t.lastIndexOf(KEY2OBJ(T)),
     "Error (" + level + ", " + seed + "): polymorphic lastIndexOf() divergence for " + T + "  (" + m.lastIndexOf(T) + ", "
+ t.lastIndexOf(KEY2OBJ(T)) + ")");
  }
```

```
  /* Now we check cloning. */

  if (level == 0) {
   ensure(m.equals(((BIG_ARRAY_BIG_LIST)m).clone()), "Error (" + level + ", " + seed + "): m does not equal
m.clone()");
   ensure(((BIG_ARRAY_BIG_LIST)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not
equal m");
  }

  /* Now we play with constructors. */
  ensure(m.equals(new BIG_ARRAY_BIG_LIST((COLLECTION)m)), "Error (" + level + ", " + seed + "): m does
not equal new (type-specific Collection m)");
  ensure((new BIG_ARRAY_BIG_LIST((COLLECTION)m)).equals(m),
 "Error (" + level + ", " + seed + "): new (type-specific nCollection m) does not equal m");
  ensure(m.equals(new BIG_ARRAY_BIG_LIST((BIG_LIST)m)), "Error (" + level + ", " + seed + "): m does not
equal new (type-specific List m)");
  ensure((new BIG_ARRAY_BIG_LIST((BIG_LIST)m)).equals(m), "Error (" + level + ", " + seed + "): new (type-
specific List m) does not equal m");
  ensure(m.equals(new BIG_ARRAY_BIG_LIST(m.listIterator())), "Error (" + level + ", " + seed + "): m does not
equal new (m.listIterator())");
  ensure((new BIG_ARRAY_BIG_LIST(m.listIterator())).equals(m), "Error (" + level + ", " + seed + "): new
(m.listIterator()) does not equal m");
  ensure(m.equals(new BIG_ARRAY_BIG_LIST(m.iterator())), "Error (" + level + ", " + seed + "): m does not equal
new (m.type_specific_iterator())");
  ensure((new BIG_ARRAY_BIG_LIST(m.iterator())).equals(m), "Error (" + level + ", " + seed + "): new
(m.type_specific_iterator()) does not equal m");


  int h = m.hashCode();

  /*
 Now we save and read m. */

  BIG_LIST m2 = null;

  try {
   java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
   java.io.OutputStream os = new java.io.FileOutputStream(ff);
   java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

   oos.writeObject(m);
   oos.close();

   java.io.InputStream is = new java.io.FileInputStream(ff);
   java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

   m2 = (BIG_LIST)ois.readObject();
```

```
  ois.close();
  ff.delete();
 }
 catch(Exception e) {
  e.printStackTrace();
  System.exit(1);
 }

#if ! KEY_CLASS_Reference
 ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

 /* Now we check that m2 actually holds that data. */

 ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
 ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
 /*
 Now we take out of m everything, and check that it is empty. */

 for(Iterator i=t.iterator(); i.hasNext();) m2.remove(i.next());

 ensure(m2.isEmpty(), "Error (" + level + ", " + seed + "): m2 is not empty (as it should be)");
#endif

 /* Now we play with iterators. */

 {
  KEY_BIG_LIST_ITERATOR i;
  KEY_BIG_LIST_ITERATOR j;
  Object J;
  i = m.listIterator();
  j = t.listIterator();

  for(int k = 0; k < 2*n; k++) {
   ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext()");
   ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious()");

   if (r.nextFloat() < .8 && i.hasNext()) {
    ensure(i.next().equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next()");

    if (r.nextFloat() < 0.2) {
     i.remove();
     j.remove();
    }
    else if (r.nextFloat() < 0.2) {
     KEY_TYPE T = genKey();
     i.set(T);
     j.set(KEY2OBJ(T));
    }
```

```
    else
if (r.nextFloat() < 0.2) {
     KEY_TYPE T = genKey();
     i.add(T);
     j.add(KEY2OBJ(T));
    }
   }
   else if (r.nextFloat() < .2 && i.hasPrevious()) {
    ensure(i.previous().equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous()");

    if (r.nextFloat() < 0.2) {
     i.remove();
     j.remove();
    }
    else if (r.nextFloat() < 0.2) {
     KEY_TYPE T = genKey();
     i.set(T);
     j.set(KEY2OBJ(T));
    }
    else if (r.nextFloat() < 0.2) {
     KEY_TYPE T = genKey();
     i.add(T);
     j.add(KEY2OBJ(T));
    }
   }

   ensure(i.nextIndex() == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
   ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

  }

 }

 {
  Object previous = null;
  Object I, J;
  long from = m.isEmpty() ? 0 : (r.nextLong() & 0x7FFFFFFFFFFFFFFL) % m.size64();
  KEY_BIG_LIST_ITERATOR
i;
  KEY_BIG_LIST_ITERATOR j;
  i = m.listIterator(from);
  j = t.listIterator(from);

  for(int k = 0; k < 2*n; k++) {
   ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting
point " + from + ")");
   ensure(i.hasPrevious() == j.hasPrevious() , "Error (" + level + ", " + seed + "): divergence in hasPrevious() (iterator
with starting point " + from + ")");
```

```
    if (r.nextFloat() < .8 && i.hasNext()) {
     ensure((I = i.next()).equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next() (" + I + ", " + J + ",
iterator with starting point " + from + ")");
     //System.err.println("Done next " + I + " " + J + "  " + badPrevious);

     if (r.nextFloat() < 0.2) {
      //System.err.println("Removing in next");
      i.remove();
      j.remove();
     }
     else if (r.nextFloat() < 0.2) {
      KEY_TYPE T = genKey();
      i.set(T);
      j.set(KEY2OBJ(T));
     }
     else if (r.nextFloat()
 < 0.2) {
      KEY_TYPE T = genKey();
      i.add(T);
      j.add(KEY2OBJ(T));
     }
    }
    else if (r.nextFloat() < .2 && i.hasPrevious()) {
     ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

     if (r.nextFloat() < 0.2) {
      //System.err.println("Removing in prev");
      i.remove();
      j.remove();
     }
     else if (r.nextFloat() < 0.2) {
      KEY_TYPE T = genKey();
      i.set(T);
      j.set(KEY2OBJ(T));
     }
     else if (r.nextFloat() < 0.2) {
      KEY_TYPE T = genKey();
      i.add(T);
      j.add(KEY2OBJ(T));
     }
    }
   }

  }

  /* Now we check that m actually holds that data. */
```

```
  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after iteration");

  /* Now we select a pair of keys and create a subset.
 */

  if (! m.isEmpty()) {
   long start = (r.nextLong() & 0x7FFFFFFFFFFFFFFL) % m.size64();
   long end = start + (r.nextLong() & 0x7FFFFFFFFFFFFFFL) % (m.size64() - start);
   //System.err.println("Checking subList from " + start + " to " + end + " (level=" + (level+1) + ")...");
   testLists(m.subList(start, end), t.subList(start, end), n, level + 1);

   ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after subList");
   ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subList");

  }

  m.clear();
  t.clear();
  ensure(m.isEmpty(), "Error (" + level + ", " + seed + "): m is not empty after clear()");
 }


 protected static void runTest(int n) {
  BIG_ARRAY_BIG_LIST m = new BIG_ARRAY_BIG_LIST();
  BIG_LIST t = BIG_LISTS.asBigList(new ARRAY_LIST());
  topList = m;
  k = new Object[n];
  nk = new Object[n];
  kt = new KEY_TYPE[n];
  nkt = new KEY_TYPE[n];

  for(int i = 0; i < n; i++) {
#if KEYS_REFERENCE
   k[i] = kt[i]
 = genKey();
   nk[i] = nkt[i] = genKey();
#else
   k[i] = new KEY_CLASS(kt[i] = genKey());
   nk[i] = new KEY_CLASS(nkt[i] = genKey());
#endif
  }

  /* We add pairs to t. */
#if KEYS_PRIMITIVE
  for(int i = 0; i < n;  i++) t.add((KEY_GENERIC_CLASS)k[i]);
#else
```

```
  for(int i = 0; i < n;  i++) t.add(k[i]);
#endif

  /* We add to m the same data */
  m.addAll(t);

  testLists(m, t, n, 0);

  System.out.println("Test OK");
  return;
 }


 public static void main(String args[]) {
  int n  = Integer.parseInt(args[1]);
  if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));


  try {
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
   else if ("test".equals(args[0])) runTest(n);
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed: " + seed);
  }
 }

#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/BigArrayBigList.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

/**  An abstract class facilitating the creation of type-specific {@linkplain it.unimi.dsi.fastutil.BidirectionalIterator
bidirectional iterators}.
*
* @deprecated As of fastutil 8 this class is no longer necessary, as its previous abstract
* methods are now default methods of the type-specific interface.
 */

@Deprecated
public abstract class KEY_ABSTRACT_BIDI_ITERATOR KEY_GENERIC extends
KEY_ABSTRACT_ITERATOR KEY_GENERIC implements KEY_BIDI_ITERATOR KEY_GENERIC {
 protected KEY_ABSTRACT_BIDI_ITERATOR() {}
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractBidirectionalIterator.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.Collection;
import java.util.Comparator;
import java.util.Iterator;
import java.util.SortedSet;
```

```
import java.util.NoSuchElementException;

/** A type-specific red-black tree set with a fast, small-footprint implementation.
*
* <p>The iterators provided by this class are type-specific
 {@link
* it.unimi.dsi.fastutil.BidirectionalIterator bidirectional iterators}.
* Moreover, the iterator returned by {@code iterator()} can be safely cast
* to a type-specific {@linkplain java.util.ListIterator list iterator}.
*/

public class RB_TREE_SET KEY_GENERIC extends ABSTRACT_SORTED_SET KEY_GENERIC implements
java.io.Serializable, Cloneable, SORTED_SET KEY_GENERIC {

 /** A reference to the root entry. */
 protected transient Entry KEY_GENERIC tree;

 /** Number of elements in this set. */
 protected int count;

 /** The entry of the first element of this set. */
 protected transient Entry KEY_GENERIC firstEntry;

 /** The entry of the last element of this set. */
 protected transient Entry KEY_GENERIC lastEntry;

 /** This set's comparator, as provided in the constructor. */
 protected Comparator<? super KEY_GENERIC_CLASS> storedComparator;

 /** This set's actual comparator; it may differ from {@link #storedComparator} because it is
  always a type-specific comparator, so
 it could be derived from the former by wrapping. */
 protected transient KEY_COMPARATOR KEY_SUPER_GENERIC actualComparator;

 private static final long serialVersionUID = -7046029254386353130L;

 {
  allocatePaths();
 }

 /** Creates a new empty tree set.
  */

 public RB_TREE_SET() {
  tree = null;
  count = 0;
 }
```

```
/** Generates the comparator that will be actually used.
 *
 * <p>When a given {@link Comparator} is specified and stored in {@link
 * #storedComparator}, we must check whether it is type-specific.  If it is
 * so, we can used directly, and we store it in {@link #actualComparator}. Otherwise,
 * we adapt it using a helper static method.
 */
private void setActualComparator() {
#if KEY_CLASS_Object
  actualComparator = storedComparator;
#else
  actualComparator = COMPARATORS.AS_KEY_COMPARATOR(storedComparator);
#endif
}


/** Creates a new empty tree set with the given comparator.
 *
 * @param c a {@link Comparator} (even better, a type-specific comparator).

 */

public RB_TREE_SET(final Comparator<? super KEY_GENERIC_CLASS> c) {
 this();
 storedComparator = c;
 setActualComparator();
}


/** Creates a new tree set copying a given collection.
 *
 * @param c a collection to be copied into the new tree set.
 */

public RB_TREE_SET(final Collection<? extends KEY_GENERIC_CLASS> c) {
 this();
 addAll(c);
}

/** Creates a new tree set copying a given sorted set (and its {@link Comparator}).
 *
 * @param s a {@link SortedSet} to be copied into the new tree set.
 */

public RB_TREE_SET(final SortedSet<KEY_GENERIC_CLASS> s) {
 this(s.comparator());
 addAll(s);
}
```

```
/** Creates a new tree set copying a given type-specific collection.
 *
 * @param c a type-specific collection to be copied into the new tree set.
 */

public RB_TREE_SET(final COLLECTION KEY_EXTENDS_GENERIC c) {
 this();
 addAll(c);
}

/** Creates a new tree set copying a given type-specific sorted set (and its {@link Comparator}).
 *
 * @param s a type-specific
sorted set to be copied into the new tree set.
 */

public RB_TREE_SET(final SORTED_SET KEY_GENERIC s) {
 this(s.comparator());
 addAll(s);
}


/** Creates a new tree set using elements provided by a type-specific iterator.
 *
 * @param i a type-specific iterator whose elements will fill the set.
 */

public RB_TREE_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i) {
 while(i.hasNext()) add(i.NEXT_KEY());
}


#if KEYS_PRIMITIVE

/** Creates a new tree set using elements provided by an iterator.
 *
 * @param i an iterator whose elements will fill the set.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public RB_TREE_SET(final Iterator<?> i) {
 this(ITERATORS.AS_KEY_ITERATOR(i));
}

#endif

/** Creates a new tree set and fills it with the elements of a given array using a given {@link Comparator}.
```

```
 *
 * @param a an array whose elements will be used to fill the set.
 * @param offset the first element to use.
 * @param length the number of elements to use.
 * @param c
a {@link Comparator} (even better, a type-specific comparator).
 */

public RB_TREE_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length, final Comparator<? super
KEY_GENERIC_CLASS> c) {
 this(c);
 ARRAYS.ensureOffsetLength(a, offset, length);
 for(int i = 0; i < length; i++) add(a[offset + i]);
}


/** Creates a new tree set and fills it with the elements of a given array.
 *
 * @param a an array whose elements will be used to fill the set.
 * @param offset the first element to use.
 * @param length the number of elements to use.
 */

public RB_TREE_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length) {
 this(a, offset, length, null);
}


/** Creates a new tree set copying the elements of an array.
 *
 * @param a an array to be copied into the new tree set.
 */

public RB_TREE_SET(final KEY_GENERIC_TYPE[] a) {
 this();
 int i = a.length;
 while(i-- != 0) add(a[i]);
}


/** Creates a new tree set copying the elements of an array using
a given {@link Comparator}.
 *
 * @param a an array to be copied into the new tree set.
 * @param c a {@link Comparator} (even better, a type-specific comparator).
 */

public RB_TREE_SET(final KEY_GENERIC_TYPE[] a, final Comparator<? super KEY_GENERIC_CLASS> c)
```

```
{
 this(c);
 int i = a.length;
 while(i-- != 0) add(a[i]);
}




/*
 * The following methods implements some basic building blocks used by
 * all accessors.  They are (and should be maintained) identical to those used in RBTreeMap.drv.
 *
 * The add()/remove() code is derived from Ben Pfaff's GNU libavl
 * (http://www.msu.edu/~pfaffben/avl/). If you want to understand what's
 * going on, you should have a look at the literate code contained therein
 * first.
 */



/** Compares two keys in the right way.
 *
 * <p>This method uses the {@link #actualComparator} if it is non-{@code null}.
 * Otherwise, it resorts to primitive type comparisons or to {@link Comparable#compareTo(Object) compareTo()}.
 *
 * @param
k1 the first key.
 * @param k2 the second key.
 * @return a number smaller than, equal to or greater than 0, as usual
 * (i.e., when k1 &lt; k2, k1 = k2 or k1 &gt; k2, respectively).
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
final int compare(final KEY_GENERIC_TYPE k1, final KEY_GENERIC_TYPE k2) {
 return actualComparator == null ? KEY_CMP(k1, k2) : actualComparator.compare(k1, k2);
}



/** Returns the entry corresponding to the given key, if it is in the tree; {@code null}, otherwise.
 *
 * @param k the key to search for.
 * @return the corresponding entry, or {@code null} if no entry with the given key exists.
 */

private Entry KEY_GENERIC findKey(final KEY_GENERIC_TYPE k) {
 Entry KEY_GENERIC e = tree;
 int cmp;
```

```
 while (e != null && (cmp = compare(k, e.key)) != 0)
  e = cmp < 0 ? e.left() : e.right();

 return e;
}

/** Locates a key.
 *
 * @param k a key.
 * @return the last entry on a search for the given key; this will be
 * the given key, if it present;
otherwise, it will be either the smallest greater key or the greatest smaller key.
 */

final Entry KEY_GENERIC locateKey(final KEY_GENERIC_TYPE k) {
 Entry KEY_GENERIC e = tree, last = tree;
 int cmp = 0;

 while (e != null && (cmp = compare(k, e.key)) != 0) {
  last = e;
  e = cmp < 0 ? e.left() : e.right();
 }

 return cmp == 0 ? e : last;
}

/** This vector remembers the path and the direction followed during the
 *  current insertion. It suffices for about 2<sup>32</sup> entries. */
private transient boolean dirPath[];
private transient Entry KEY_GENERIC nodePath[];

SUPPRESS_WARNINGS_KEY_UNCHECKED_RAWTYPES
private void allocatePaths() {
 dirPath = new boolean[64];
 nodePath = new Entry[64];
}

@Override
public boolean add(final KEY_GENERIC_TYPE k) {
 int maxDepth = 0;

 if (tree == null) { // The case of the empty tree is treated separately.
  count++;
  tree = lastEntry = firstEntry = new Entry KEY_GENERIC_DIAMOND(k);
 }
 else {
  Entry KEY_GENERIC
```

```
p = tree, e;
  int cmp, i = 0;

  while(true) {
   if ((cmp = compare(k, p.key)) == 0) {
    // We clean up the node path, or we could have stale references later.
    while(i-- != 0) nodePath[i] = null;
    return false;
   }

   nodePath[i] = p;

   if (dirPath[i++] = cmp > 0) {
    if (p.succ()) {
     count++;
     e = new Entry KEY_GENERIC_DIAMOND(k);

     if (p.right == null) lastEntry = e;

     e.left = p;
     e.right = p.right;

     p.right(e);

     break;
    }

    p = p.right;
   }
   else {
    if (p.pred()) {
     count++;
     e = new Entry KEY_GENERIC_DIAMOND(k);

     if (p.left == null) firstEntry = e;

     e.right = p;
     e.left = p.left;

     p.left(e);

     break;
    }

    p = p.left;
   }
  }
```

```
  maxDepth = i--;

  while(i > 0 && ! nodePath[i].black()) {
   if (! dirPath[i - 1]) {
    Entry KEY_GENERIC y = nodePath[i - 1].right;

    if (! nodePath[i - 1].succ() && ! y.black()) {
     nodePath[i].black(true);
     y.black(true);
     nodePath[i
- 1].black(false);
     i -= 2;
    }
    else {
     Entry KEY_GENERIC x;

     if (! dirPath[i]) y = nodePath[i];
     else {
      x = nodePath[i];
      y = x.right;
      x.right = y.left;
      y.left = x;
      nodePath[i - 1].left = y;

      if (y.pred()) {
       y.pred(false);
       x.succ(y);
      }
     }

     x = nodePath[i - 1];
     x.black(false);
     y.black(true);

     x.left = y.right;
     y.right = x;
     if (i < 2) tree = y;
     else {
      if (dirPath[i - 2]) nodePath[i - 2].right = y;
      else nodePath[i - 2].left = y;
     }

     if (y.succ()) {
      y.succ(false);
      x.pred(y);
     }
     break;
    }
```

```
    }
    else {
     Entry KEY_GENERIC y = nodePath[i - 1].left;

     if (! nodePath[i - 1].pred() && ! y.black()) {
      nodePath[i].black(true);
      y.black(true);
      nodePath[i - 1].black(false);
      i -= 2;
     }
     else {
      Entry KEY_GENERIC x;

      if (dirPath[i])
y = nodePath[i];
      else {
       x = nodePath[i];
       y = x.left;
       x.left = y.right;
       y.right = x;
       nodePath[i - 1].right = y;

       if (y.succ()) {
        y.succ(false);
        x.pred(y);
       }

      }

      x = nodePath[i - 1];
      x.black(false);
      y.black(true);

      x.right = y.left;
      y.left = x;
      if (i < 2) tree = y;
      else {
       if (dirPath[i - 2]) nodePath[i - 2].right = y;
       else nodePath[i - 2].left = y;
      }

      if (y.pred()){
       y.pred(false);
       x.succ(y);
      }

      break;
     }
```

```
  }
 }
}
tree.black(true);
// We clean up the node path, or we could have stale references later.
while(maxDepth-- != 0) nodePath[maxDepth] = null;
return true;
}


SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public boolean remove(final KEY_TYPE k) {
 if (tree == null) return false;

 Entry KEY_GENERIC p = tree;
 int cmp;
 int i = 0;
 final KEY_GENERIC_TYPE kk = KEY_GENERIC_CAST k;

 while(true)
{
  if ((cmp = compare(kk, p.key)) == 0) break;

  dirPath[i] = cmp > 0;
  nodePath[i] = p;

  if (dirPath[i++]) {
   if ((p = p.right()) == null) {
    // We clean up the node path, or we could have stale references later.
    while(i-- != 0) nodePath[i] = null;
    return false;
   }
  }
  else {
   if ((p = p.left()) == null) {
    // We clean up the node path, or we could have stale references later.
    while(i-- != 0) nodePath[i] = null;
    return false;
   }
  }
 }

 if (p.left == null) firstEntry = p.next();
 if (p.right == null) lastEntry = p.prev();

 if (p.succ()) {
  if (p.pred()) {
```

```
  if (i == 0) tree = p.left;
  else {
   if (dirPath[i - 1]) nodePath[i - 1].succ(p.right);
   else nodePath[i - 1].pred(p.left);
  }
 }
 else {
  p.prev().right = p.right;

  if (i == 0) tree = p.left;
  else {
   if (dirPath[i - 1]) nodePath[i - 1].right = p.left;
   else nodePath[i - 1].left = p.left;
  }
 }
}
else {
 boolean
color;
 Entry KEY_GENERIC r = p.right;

 if (r.pred()) {
  r.left = p.left;
  r.pred(p.pred());
  if (! r.pred()) r.prev().right = r;
  if (i == 0) tree = r;
  else {
   if (dirPath[i - 1]) nodePath[i - 1].right = r;
   else nodePath[i - 1].left = r;
  }

  color = r.black();
  r.black(p.black());
  p.black(color);
  dirPath[i] = true;
  nodePath[i++] = r;
 }
 else {
  Entry KEY_GENERIC s;
  int j = i++;

  while(true) {
   dirPath[i] = false;
   nodePath[i++] = r;
   s = r.left;
   if (s.pred()) break;
   r = s;
  }
```

```
    dirPath[j] = true;
    nodePath[j] = s;

    if (s.succ()) r.pred(s);
    else r.left = s.right;

    s.left = p.left;

    if (! p.pred()) {
     p.prev().right =  s;
     s.pred(false);
    }

    s.right(p.right);

    color = s.black();
    s.black(p.black());
    p.black(color);

    if (j == 0) tree = s;
    else {
     if (dirPath[j - 1]) nodePath[j - 1].right = s;
     else nodePath[j - 1].left = s;
    }
   }
  }

  int
 maxDepth = i;

  if (p.black()) {
   for(; i > 0; i--) {
    if (dirPath[i - 1] && ! nodePath[i - 1].succ() ||
      ! dirPath[i - 1] && ! nodePath[i - 1].pred()) {
     Entry KEY_GENERIC x = dirPath[i - 1] ? nodePath[i - 1].right : nodePath[i - 1].left;

     if (! x.black()) {
      x.black(true);
      break;
     }
    }

    if (! dirPath[i - 1]) {
     Entry KEY_GENERIC w = nodePath[i - 1].right;

     if (! w.black()) {
      w.black(true);
```

```
    nodePath[i - 1].black(false);

    nodePath[i - 1].right = w.left;
    w.left = nodePath[i - 1];

    if (i < 2) tree = w;
    else {
     if (dirPath[i - 2]) nodePath[i - 2].right = w;
     else nodePath[i - 2].left = w;
    }

    nodePath[i] = nodePath[i - 1];
    dirPath[i] = false;
    nodePath[i - 1] = w;
    if (maxDepth == i++) maxDepth++;

    w = nodePath[i - 1].right;
   }

   if ((w.pred() || w.left.black()) &&
    (w.succ() || w.right.black())) {
    w.black(false);
   }
   else
{
    if (w.succ() || w.right.black()) {
     Entry KEY_GENERIC y = w.left;

     y.black (true);
     w.black(false);
     w.left = y.right;
     y.right = w;
     w = nodePath[i - 1].right = y;

     if (w.succ()) {
      w.succ(false);
      w.right.pred(w);
     }
    }

    w.black(nodePath[i - 1].black());
    nodePath[i - 1].black(true);
    w.right.black(true);

    nodePath[i - 1].right = w.left;
    w.left = nodePath[i - 1];

    if (i < 2) tree = w;
```

```
      else {
       if (dirPath[i - 2]) nodePath[i - 2].right = w;
       else nodePath[i - 2].left = w;
      }

      if (w.pred()) {
       w.pred(false);
       nodePath[i - 1].succ(w);
      }
      break;
     }
    }
    else {
     Entry KEY_GENERIC w = nodePath[i - 1].left;

     if (! w.black()) {
      w.black (true);
      nodePath[i - 1].black(false);

      nodePath[i - 1].left = w.right;
      w.right = nodePath[i - 1];

      if (i < 2) tree = w;
      else {
       if (dirPath[i
- 2]) nodePath[i - 2].right = w;
       else nodePath[i - 2].left = w;
      }

      nodePath[i] = nodePath[i - 1];
      dirPath[i] = true;
      nodePath[i - 1] = w;
      if (maxDepth == i++) maxDepth++;

      w = nodePath[i - 1].left;
     }

     if ((w.pred() || w.left.black()) &&
       (w.succ() || w.right.black())) {
      w.black(false);
     }
     else {
      if (w.pred() || w.left.black()) {
       Entry KEY_GENERIC y = w.right;

       y.black(true);
       w.black (false);
       w.right = y.left;
```

```
      y.left = w;
      w = nodePath[i - 1].left = y;

      if (w.pred()) {
       w.pred(false);
       w.left.succ(w);
      }
     }

     w.black(nodePath[i - 1].black());
     nodePath[i - 1].black(true);
     w.left.black(true);

     nodePath[i - 1].left = w.right;
     w.right = nodePath[i - 1];

     if (i < 2) tree = w;
     else {
      if (dirPath[i - 2]) nodePath[i - 2].right = w;
      else nodePath[i - 2].left = w;
     }

     if (w.succ())
    {
      w.succ(false);
      nodePath[i - 1].pred(w);
     }
     break;
    }
   }
  }

  if (tree != null) tree.black(true);
 }

 count--;
 // We clean up the node path, or we could have stale references later.
 while(maxDepth-- != 0) nodePath[maxDepth] = null;
 return true;
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public boolean contains(final KEY_TYPE k) {
 return findKey(KEY_GENERIC_CAST k) != null;
}

#if KEY_CLASS_Object
```

```
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public K get(final KEY_TYPE k) {
  final Entry KEY_GENERIC entry = findKey(KEY_GENERIC_CAST k);
  return entry == null ? null : entry.key;
 }
#endif

 @Override
 public void clear() {
  count = 0;
  tree = null;
  firstEntry = lastEntry = null;
 }



 /** This class represent an entry in a tree set.
  *
  * <p>We use the only "metadata", i.e., {@link Entry#info}, to store
  * information about color, predecessor status and successor status.
  *
  * <p>Note that since the class is recursive, it can be

 * considered equivalently a tree.
  */

 private static final class Entry KEY_GENERIC implements Cloneable {
  /** The the bit in this mask is true, the node is black. */
  private static final int BLACK_MASK = 1;
  /** If the bit in this mask is true, {@link #right} points to a successor. */
  private static final int SUCC_MASK = 1 << 31;
  /** If the bit in this mask is true, {@link #left} points to a predecessor. */
  private static final int PRED_MASK = 1 << 30;
  /** The key of this entry. */
  KEY_GENERIC_TYPE key;
  /** The pointers to the left and right subtrees. */
  Entry KEY_GENERIC left, right;
  /** This integers holds different information in different bits (see {@link #SUCC_MASK}, {@link
#PRED_MASK} and {@link #BLACK_MASK}). */
  int info;

  Entry() {}

  /** Creates a new red entry with the given key.
   *
   * @param k a key.
   */
  Entry(final KEY_GENERIC_TYPE k) {
   this.key = k;
```

```
 info = SUCC_MASK | PRED_MASK;
}

/** Returns the left subtree.
 *
 * @return
the left subtree ({@code null} if the left
 * subtree is empty).
 */
Entry KEY_GENERIC left() {
 return (info & PRED_MASK) != 0 ? null : left;
}

/** Returns the right subtree.
 *
 * @return the right subtree ({@code null} if the right
 * subtree is empty).
 */
Entry KEY_GENERIC right() {
 return (info & SUCC_MASK) != 0 ? null : right;
}

/** Checks whether the left pointer is really a predecessor.
 * @return true if the left pointer is a predecessor.
 */
boolean pred() {
 return (info & PRED_MASK) != 0;
}

/** Checks whether the right pointer is really a successor.
 * @return true if the right pointer is a successor.
 */
boolean succ() {
 return (info & SUCC_MASK) != 0;
}

/** Sets whether the left pointer is really a predecessor.
 * @param pred if true then the left pointer will be considered a predecessor.
 */
void pred(final boolean pred) {
 if (pred) info |= PRED_MASK;
 else info &= ~PRED_MASK;
}

/** Sets whether
the right pointer is really a successor.
 * @param succ if true then the right pointer will be considered a successor.
 */
```

```
void succ(final boolean succ) {
 if (succ) info |= SUCC_MASK;
 else info &= ~SUCC_MASK;
}

/** Sets the left pointer to a predecessor.
 * @param pred the predecessr.
 */
void pred(final Entry KEY_GENERIC pred) {
 info |= PRED_MASK;
 left = pred;
}

/** Sets the right pointer to a successor.
 * @param succ the successor.
 */
void succ(final Entry KEY_GENERIC succ) {
 info |= SUCC_MASK;
 right = succ;
}

/** Sets the left pointer to the given subtree.
 * @param left the new left subtree.
 */
void left(final Entry KEY_GENERIC left) {
 info &= ~PRED_MASK;
 this.left = left;
}

/** Sets the right pointer to the given subtree.
 * @param right the new right subtree.
 */
void right(final Entry KEY_GENERIC right) {
 info &= ~SUCC_MASK;
 this.right = right;
}

/** Returns whether this node is black.

* @return true iff this node is black.
 */
boolean black() {
 return (info & BLACK_MASK) != 0;
}

/** Sets whether this node is black.
 * @param black if true, then this node becomes black; otherwise, it becomes red..
 */
```

```
void black(final boolean black) {
 if (black) info |= BLACK_MASK;
 else info &= ~BLACK_MASK;
}

/** Computes the next entry in the set order.
 *
 * @return the next entry ({@code null}) if this is the last entry).
 */

Entry KEY_GENERIC next() {
 Entry KEY_GENERIC next = this.right;
 if ((info & SUCC_MASK) == 0) while ((next.info & PRED_MASK) == 0) next = next.left;
 return next;
}

/** Computes the previous entry in the set order.
 *
 * @return the previous entry ({@code null}) if this is the first entry).
 */

Entry KEY_GENERIC prev() {
 Entry KEY_GENERIC prev = this.left;
 if ((info & PRED_MASK) == 0) while ((prev.info & SUCC_MASK) == 0) prev = prev.right;
 return prev;
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public
Entry KEY_GENERIC clone() {
 Entry KEY_GENERIC c;
 try {
  c = (Entry KEY_GENERIC)super.clone();
 }
 catch(CloneNotSupportedException cantHappen) {
  throw new InternalError();
 }

 c.key = key;
 c.info = info;

 return c;
}

public boolean equals(final Object o) {
 if (!(o instanceof Entry)) return false;
 Entry KEY_GENERIC_WILDCARD e = (Entry KEY_GENERIC_WILDCARD)o;
```

```
 return KEY_EQUALS(key, e.key);
 }

 public int hashCode() {
  return KEY2JAVAHASH_NOT_NULL(key);
 }


 public String toString() {
  return String.valueOf(key);
 }

 /*
   public void prettyPrint() {
   prettyPrint(0);
   }


   public void prettyPrint(int level) {
   if (pred()) {
   for (int i = 0; i < level; i++)
   System.err.print("  ");
   System.err.println("pred: " + left);
   }
   else if (left != null)
   left.prettyPrint(level +1);
   for (int i = 0; i < level; i++)
   System.err.print("  ");
   System.err.println(key + " (" + (black()
? "black" : "red") + ")");
   if (succ()) {
   for (int i = 0; i < level; i++)
   System.err.print("  ");
   System.err.println("succ: " + right);
   }
   else if (right != null)
   right.prettyPrint(level + 1);
   }*/
 }

 /*
  public void prettyPrint() {
  System.err.println("size: " + count);
  if (tree != null) tree.prettyPrint();
  }
 */
```

```
@Override
public int size() {
 return count;
}

@Override
public boolean isEmpty() {
 return count == 0;
}

@Override
public KEY_GENERIC_TYPE FIRST() {
 if (tree == null) throw new NoSuchElementException();
 return firstEntry.key;
}

@Override
public KEY_GENERIC_TYPE LAST() {
 if (tree == null) throw new NoSuchElementException();
 return lastEntry.key;
}


/** An iterator on the whole range.
 *
 * <p>This class can iterate in both directions on a threaded tree.
 */

private class SetIterator implements KEY_LIST_ITERATOR KEY_GENERIC {
 /** The entry that will be returned by the next call to {@link java.util.ListIterator#previous()}
(or {@code null} if no previous entry exists). */
 Entry KEY_GENERIC prev;
 /** The entry that will be returned by the next call to {@link java.util.ListIterator#next()} (or {@code null} if no
next entry exists). */
 Entry KEY_GENERIC next;
 /** The last entry that was returned (or {@code null} if we did not iterate or used {@link #remove()}). */
 Entry KEY_GENERIC curr;
 /** The current index (in the sense of a {@link java.util.ListIterator}). Note that this value is not meaningful when
this iterator has been created using the nonempty constructor.*/
 int index = 0;

 SetIterator() {
  next = firstEntry;
 }

 SetIterator(final KEY_GENERIC_TYPE k) {
  if ((next = locateKey(k)) != null) {
   if (compare(next.key, k) <= 0) {
```

```
    prev = next;
    next = next.next();
   }
   else prev = next.prev();
  }
 }

 @Override
 public boolean hasNext() { return next != null; }
 @Override
 public boolean hasPrevious() { return prev != null; }

 void updateNext()
 { next = next.next(); }
 void updatePrevious() { prev = prev.prev(); }

 @Override
 public KEY_GENERIC_TYPE NEXT_KEY() { return nextEntry().key; }
 @Override
 public KEY_GENERIC_TYPE PREV_KEY() { return previousEntry().key; }

 Entry KEY_GENERIC nextEntry() {
  if (! hasNext()) throw new NoSuchElementException();
  curr = prev = next;
  index++;
  updateNext();
  return curr;
 }

 Entry KEY_GENERIC previousEntry() {
  if (! hasPrevious()) throw new NoSuchElementException();
  curr = next = prev;
  index--;
  updatePrevious();
  return curr;
 }

 @Override
 public int nextIndex() { return index; }
 @Override
 public int previousIndex() { return index - 1; }

 @Override
 public void remove() {
  if (curr == null) throw new IllegalStateException();
  /* If the last operation was a next(), we are removing an entry that preceeds
     the current index, and thus we must decrement it. */
  if (curr == prev) index--;
```

```
  next = prev = curr;
  updatePrevious();
  updateNext();
  RB_TREE_SET.this.remove(curr.key);
  curr
 = null;
  }
 }

 @Override
 public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new SetIterator(); }

 @Override
 public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
SetIterator(from); }

 @Override
 public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return actualComparator; }

 @Override
 public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) { return new
Subset(KEY_NULL, true, to, false); }

 @Override
 public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) { return new Subset(from,
false, KEY_NULL, true); }

 @Override
 public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_TYPE from, final KEY_GENERIC_TYPE
to) { return new Subset(from, false, to, false); }

 /** A subset with given range.
  *
  * <p>This class represents a subset. One has to specify the left/right
  * limits (which can be set to -&infin; or &infin;). Since the subset is a
  * view on
 the set, at a given moment it could happen that the limits of
  * the range are not any longer in the main set. Thus, things such as
  * {@link java.util.SortedSet#first()} or {@link java.util.Collection#size()} must be always computed
  * on-the-fly.
  */
 private final class Subset extends ABSTRACT_SORTED_SET KEY_GENERIC implements java.io.Serializable,
SORTED_SET KEY_GENERIC {
  private static final long serialVersionUID = -7046029254386353129L;

  /** The start of the subset range, unless {@link #bottom} is true. */
  KEY_GENERIC_TYPE from;
  /** The end of the subset range, unless {@link #top} is true. */
```

```
 KEY_GENERIC_TYPE to;
 /** If true, the subset range starts from -&infin;. */
 boolean bottom;
 /** If true, the subset range goes to &infin;. */
 boolean top;

 /** Creates a new subset with given key range.
  *
  * @param from the start of the subset range.
  * @param bottom if true, the first parameter is ignored and the range starts from -&infin;.
  * @param to the
end of the subset range.
  * @param top if true, the third parameter is ignored and the range goes to &infin;.
  */
 public Subset(final KEY_GENERIC_TYPE from, final boolean bottom, final KEY_GENERIC_TYPE to, final
boolean top) {
  if (! bottom && ! top && RB_TREE_SET.this.compare(from, to) > 0) throw new IllegalArgumentException("Start
element (" + from  + ") is larger than end element (" + to + ")");

  this.from = from;
  this.bottom = bottom;
  this.to = to;
  this.top = top;
 }

 @Override
 public void clear() {
  final SubsetIterator i = new SubsetIterator();
  while(i.hasNext()) {
   i.NEXT_KEY();
   i.remove();
  }
 }

 /** Checks whether a key is in the subset range.
  * @param k a key.
  * @return true if is the key is in the subset range.
  */
 final boolean in(final KEY_GENERIC_TYPE k) {
  return (bottom || RB_TREE_SET.this.compare(k, from) >= 0) &&
   (top || RB_TREE_SET.this.compare(k, to) < 0);
 }

 @Override
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public
boolean contains(final KEY_TYPE k) {
  return in(KEY_GENERIC_CAST k) && RB_TREE_SET.this.contains(k);
```

```
 }

 @Override
 public boolean add(final KEY_GENERIC_TYPE  k) {
  if (! in(k)) throw new IllegalArgumentException("Element (" + k + ") out of range [" + (bottom ? "-" :
String.valueOf(from)) + ", " + (top ? "-" : String.valueOf(to)) + ")");
  return RB_TREE_SET.this.add(k);
 }

 @Override
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public boolean remove(final KEY_TYPE k) {
  if (! in(KEY_GENERIC_CAST k)) return false;
  return RB_TREE_SET.this.remove(k);
 }

 @Override
 public int size() {
  final SubsetIterator i = new SubsetIterator();
  int n = 0;

  while(i.hasNext()) {
   n++;
   i.NEXT_KEY();
  }

  return n;
 }


 @Override
 public boolean isEmpty() {
  return ! new SubsetIterator().hasNext();
 }

 @Override
 public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() {
  return actualComparator;
 }

 @Override
 public KEY_BIDI_ITERATOR KEY_GENERIC iterator()
 {
  return new SubsetIterator();
 }

 @Override
 public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) {
```

```
 return new SubsetIterator(from);
}

@Override
public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) {
 if (top) return new Subset(from, bottom, to, false);
 return compare(to, this.to) < 0 ? new Subset(from, bottom, to, false) : this;
}

@Override
public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) {
 if (bottom) return new Subset(from, false, to, top);
 return compare(from, this.from) > 0 ? new Subset(from, false, to, top) : this;
}

@Override
public SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE to) {
 if (top && bottom) return new Subset(from, false, to, false);
 if (! top) to = compare(to, this.to) < 0 ? to : this.to;
 if (! bottom) from = compare(from, this.from) > 0 ? from : this.from;
   if (! top && ! bottom && from == this.from && to == this.to) return this;
 return
new Subset(from, false, to, false);
}


/** Locates the first entry.
 *
 * @return the first entry of this subset, or {@code null} if the subset is empty.
 */
public RB_TREE_SET.Entry KEY_GENERIC firstEntry() {
 if (tree == null) return null;
 // If this subset goes to -infinity, we return the main set first entry; otherwise, we locate the start of the set.
 RB_TREE_SET.Entry KEY_GENERIC e;
 if (bottom) e = firstEntry;
 else {
  e = locateKey(from);
  // If we find either the start or something greater we're OK.
  if (compare(e.key, from) < 0) e = e.next();
 }
 // Finally, if this subset doesn't go to infinity, we check that the resulting key isn't greater than the end.
 if (e == null || ! top && compare(e.key, to) >= 0) return null;
 return e;
}

/** Locates the last entry.
 *
 * @return the last entry of this subset, or {@code null} if the subset is empty.
```

```
  */
 public RB_TREE_SET.Entry KEY_GENERIC lastEntry() {
  if (tree == null)
return null;
  // If this subset goes to infinity, we return the main set last entry; otherwise, we locate the end of the set.
  RB_TREE_SET.Entry KEY_GENERIC e;
  if (top) e = lastEntry;
  else {
   e = locateKey(to);
   // If we find something smaller than the end we're OK.
   if (compare(e.key, to) >= 0) e = e.prev();
  }
  // Finally, if this subset doesn't go to -infinity, we check that the resulting key isn't smaller than the start.
  if (e == null || ! bottom && compare(e.key, from) < 0) return null;
  return e;
 }


 @Override
 public KEY_GENERIC_TYPE FIRST() {
  RB_TREE_SET.Entry KEY_GENERIC e = firstEntry();
  if (e == null) throw new NoSuchElementException();
  return e.key;
 }

 @Override
 public KEY_GENERIC_TYPE LAST() {
  RB_TREE_SET.Entry KEY_GENERIC e = lastEntry();
  if (e == null) throw new NoSuchElementException();
  return e.key;
 }

 /** An iterator for subranges.
  *
  * <p>This class inherits from {@link SetIterator}, but overrides
the methods that
  * update the pointer after a {@link java.util.ListIterator#next()} or {@link java.util.ListIterator#previous()}. If we
would
  * move out of the range of the subset we just overwrite the next or previous
  * entry with {@code null}.
  */
 private final class SubsetIterator extends SetIterator {
  SubsetIterator() {
   next = firstEntry();
  }

  SubsetIterator(final KEY_GENERIC_TYPE k) {
   this();
```

```
  if (next != null) {
   if (! bottom && compare(k, next.key) < 0) prev = null;
   else if (! top && compare(k, (prev = lastEntry()).key) >= 0) next = null;
   else {
    next = locateKey(k);

    if (compare(next.key, k) <= 0) {
     prev = next;
     next = next.next();
    }
    else prev = next.prev();
   }
  }
 }

 @Override
 void updatePrevious() {
  prev = prev.prev();
  if (! bottom && prev != null && RB_TREE_SET.this.compare(prev.key, from) < 0) prev = null;
 }

 @Override
 void updateNext() {
  next = next.next();
  if
(! top && next != null && RB_TREE_SET.this.compare(next.key, to) >= 0) next = null;
 }
}
}



/** Returns a deep copy of this tree set.
 *
 * <p>This method performs a deep copy of this tree set; the data stored in the
 * set, however, is not cloned. Note that this makes a difference only for object keys.
 *
 * @return a deep copy of this tree set.
 */
@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public Object clone() {
 RB_TREE_SET KEY_GENERIC c;
 try {
  c = (RB_TREE_SET KEY_GENERIC)super.clone();
 }
 catch(CloneNotSupportedException cantHappen) {
```

```
  throw new InternalError();
 }

 c.allocatePaths();

 if (count != 0) {
  // Also this apparently unfathomable code is derived from GNU libavl.
  Entry KEY_GENERIC e, p, q, rp = new Entry KEY_GENERIC_DIAMOND(), rq = new Entry
KEY_GENERIC_DIAMOND();

  p = rp;
  rp.left(tree);

  q = rq;
  rq.pred(null);

  while(true) {
   if (! p.pred()) {
    e = p.left.clone();
    e.pred(q.left);
    e.succ(q);
    q.left(e);

    p
 = p.left;
    q = q.left;
   }
   else {
    while(p.succ()) {
     p = p.right;

     if (p == null) {
      q.right = null;
      c.tree = rq.left;

      c.firstEntry = c.tree;
      while(c.firstEntry.left != null) c.firstEntry = c.firstEntry.left;
      c.lastEntry = c.tree;
      while(c.lastEntry.right != null) c.lastEntry = c.lastEntry.right;

      return c;
     }
     q = q.right;
    }

    p = p.right;
    q = q.right;
   }
```

```
    if (! p.succ()) {
     e = p.right.clone();
     e.succ(q.right);
     e.pred(q);
     q.right(e);
    }
   }
  }

  return c;
 }


 private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
  int n = count;
  SetIterator i = new SetIterator();

  s.defaultWriteObject();
  while(n-- != 0) s.WRITE_KEY(i.NEXT_KEY());
 }


 /** Reads the given number of entries from the input stream, returning the corresponding tree.
  *
  * @param s the input stream.
  * @param n the (positive) number
 of entries to read.
  * @param pred the entry containing the key that preceeds the first key in the tree.
  * @param succ the entry containing the key that follows the last key in the tree.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 private Entry KEY_GENERIC readTree(final java.io.ObjectInputStream s, final int n, final Entry KEY_GENERIC
pred, final Entry KEY_GENERIC succ) throws java.io.IOException, ClassNotFoundException {
  if (n == 1) {
   final Entry KEY_GENERIC top = new Entry KEY_GENERIC_DIAMOND(KEY_GENERIC_CAST
s.READ_KEY());
   top.pred(pred);
   top.succ(succ);
   top.black(true);

   return top;
  }

  if (n == 2) {
   /* We handle separately this case so that recursion will
    *always* be on nonempty subtrees. */
   final Entry KEY_GENERIC top = new Entry KEY_GENERIC_DIAMOND(KEY_GENERIC_CAST
```

```
s.READ_KEY());
  top.black(true);
  top.right(new Entry KEY_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY()));
  top.right.pred(top);
  top.pred(pred);
  top.right.succ(succ);

  return
 top;
  }

 // The right subtree is the largest one.
 final int rightN = n / 2, leftN = n - rightN - 1;

 final Entry KEY_GENERIC top = new Entry KEY_GENERIC_DIAMOND();

 top.left(readTree(s, leftN, pred, top));

 top.key = KEY_GENERIC_CAST s.READ_KEY();
 top.black(true);

 top.right(readTree(s, rightN, top, succ));

 if (n + 2 == ((n + 2)  & -(n + 2))) top.right.black(false); // Quick test for determining whether n + 2 is a power of 2.

 return top;
}


private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
 s.defaultReadObject();
 /* The storedComparator is now correctly set, but we must restore
    on-the-fly the actualComparator. */
 setActualComparator();
 allocatePaths();

 if (count != 0) {
  tree = readTree(s, count, null, null);
  Entry KEY_GENERIC e;

  e = tree;
  while(e.left() != null) e = e.left();
  firstEntry = e;

  e = tree;
  while(e.right() != null) e = e.right();
  lastEntry = e;
  }
```

```
 }


#ifdef
 ASSERTS_CODE
 private void checkNodePath() {
  for(int i = nodePath.length; i-- != 0;) assert nodePath[i] == null : i;
 }

 private static KEY_GENERIC int checkTree(Entry KEY_GENERIC e, int d, int D) {
  if (e == null) return 0;
  if (e.black()) d++;
  if (e.left() != null) D = checkTree(e.left(), d, D);
  if (e.right() != null) D = checkTree(e.right(), d, D);
  if (e.left() == null && e.right() == null) {
   if (D == -1) D = d;
   else if (D != d) throw new AssertionError("Mismatch between number of black nodes (" + D + " and " + d + ")");
  }
  return D;
 }
#endif

#ifdef TEST

 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#else
  return Integer.toBinaryString(r.nextInt());
#endif
 }

 private static java.text.NumberFormat format
 = new java.text.DecimalFormat("#,###.00");
 private static java.text.FieldPosition p = new java.text.FieldPosition(0);

 private static String format(double d) {
  StringBuffer s = new StringBuffer();
  return format.format(d, s, p).toString();
 }

 private static void speedTest(int n, boolean comp) {
  int i, j;
```

```
RB_TREE_SET m;
java.util.TreeSet t;
KEY_TYPE k[] = new KEY_TYPE[n];
KEY_TYPE nk[] = new KEY_TYPE[n];
long ms;

for(i = 0; i < n; i++) {
 k[i] = genKey();
 nk[i] = genKey();
}

double totAdd = 0, totYes = 0, totNo = 0, totIterFor = 0, totIterBack = 0, totRemYes = 0, d, dd;

if (comp) {
 for(j = 0; j < 20; j++) {

  t = new java.util.TreeSet();

  /* We first add all pairs to t. */
  for(i = 0; i < n;  i++) t.add(KEY2OBJ(k[i]));

  /* Then we remove the first half and put it back. */
  for(i = 0; i < n/2;  i++) t.remove(KEY2OBJ(k[i]));

  ms = System.currentTimeMillis();
  for(i = 0; i < n/2;  i++) t.add(KEY2OBJ(k[i]));
  d = System.currentTimeMillis()
- ms;

  /* Then we remove the other half and put it back again. */
  ms = System.currentTimeMillis();
  for(i = n/2; i < n;  i++) t.remove(KEY2OBJ(k[i]));
  dd = System.currentTimeMillis() - ms ;

  ms = System.currentTimeMillis();
  for(i = n/2; i < n;  i++) t.add(KEY2OBJ(k[i]));
  d += System.currentTimeMillis() - ms;
  if (j > 2) totAdd += n/d;
  System.out.print("Add: " + format(n/d) +" K/s ");

  /* Then we remove again the first half. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n/2;  i++) t.remove(KEY2OBJ(k[i]));
  dd += System.currentTimeMillis() - ms ;
  if (j > 2) totRemYes += n/dd;
  System.out.print("RemYes: " + format(n/dd) +" K/s ");

  /* And then we put it back. */
```

```
    for(i = 0; i < n/2;  i++) t.add(KEY2OBJ(k[i]));

    /* We check for pairs in t. */
    ms = System.currentTimeMillis();
    for(i = 0; i < n;  i++) t.contains(KEY2OBJ(k[i]));
    d = 1.0 * n / (System.currentTimeMillis() - ms);
    if (j > 2)
 totYes += d;
    System.out.print("Yes: " + format(d) +" K/s ");

    /* We check for pairs not in t. */
    ms = System.currentTimeMillis();
    for(i = 0; i < n;  i++) t.contains(KEY2OBJ(nk[i]));
    d = 1.0 * n / (System.currentTimeMillis() - ms);
    if (j > 2) totNo += d;
    System.out.print("No: " + format(d) +" K/s ");

    /* We iterate on t. */
    ms = System.currentTimeMillis();
    for(Iterator it = t.iterator(); it.hasNext(); it.next());
    d = 1.0 * n / (System.currentTimeMillis() - ms);
    if (j > 2) totIterFor += d;
    System.out.print("IterFor: " + format(d) +" K/s ");

    System.out.println();
   }

   System.out.println();
   System.out.println("java.util Add: " + format(totAdd/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s
Yes: " + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3)) + " K/s IterFor: " + format(totIterFor/(j-3))  + "
K/s");

   System.out.println();

   totAdd = totYes = totNo = totIterFor = totIterBack = totRemYes = 0;

  }

  for(j
 = 0; j < 20; j++) {

   m = new RB_TREE_SET();


   /* We first add all pairs to m. */
   for(i = 0; i < n;  i++) m.add(k[i]);

   /* Then we remove the first half and put it back. */
```

```
for(i = 0; i < n/2;  i++) m.remove(k[i]);

ms = System.currentTimeMillis();
for(i = 0; i < n/2;  i++) m.add(k[i]);
d = System.currentTimeMillis() - ms;

/* Then we remove the other half and put it back again. */
ms = System.currentTimeMillis();
for(i = n/2; i < n;  i++) m.remove(k[i]);
dd = System.currentTimeMillis() - ms ;

ms = System.currentTimeMillis();
for(i = n/2; i < n;  i++) m.add(k[i]);
d += System.currentTimeMillis() - ms;
if (j > 2) totAdd += n/d;
System.out.print("Add: " + format(n/d) +" K/s ");

/* Then we remove again the first half. */
ms = System.currentTimeMillis();
for(i = 0; i < n/2;  i++) m.remove(k[i]);
dd += System.currentTimeMillis() - ms ;
if (j > 2) totRemYes += n/dd;
System.out.print("RemYes: " + format(n/dd) +" K/s
");

/* And then we put it back. */
for(i = 0; i < n/2;  i++) m.add(k[i]);

/* We check for pairs in m. */
ms = System.currentTimeMillis();
for(i = 0; i < n;  i++) m.contains(k[i]);
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totYes += d;
System.out.print("Yes: " + format(d) +" K/s ");

/* We check for pairs not in m. */
ms = System.currentTimeMillis();
for(i = 0; i < n;  i++) m.contains(nk[i]);
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totNo += d;
System.out.print("No: " + format(d) +" K/s ");

/* We iterate on m. */
KEY_LIST_ITERATOR it = (KEY_LIST_ITERATOR)m.iterator();
ms = System.currentTimeMillis();
for(; it.hasNext(); it.NEXT_KEY());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIterFor += d;
```

```
  System.out.print("IterFor: " + format(d) +" K/s ");

  /* We iterate back on m. */
  ms = System.currentTimeMillis();
  for(; it.hasPrevious(); it.PREV_KEY());
  d = 1.0 * n
/ (System.currentTimeMillis() - ms);
  if (j > 2) totIterBack += d;
  System.out.print("IterBack: " + format(d) +" K/s ");

  System.out.println();
 }


 System.out.println();
 System.out.println("fastutil  Add: " + format(totAdd/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s
Yes: " + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3)) + " K/s IterFor: " + format(totIterFor/(j-3))  + " K/s
IterBack: " + format(totIterBack/(j-3)) + "K/s");

 System.out.println();
}


private static boolean valEquals(Object o1, Object o2) {
 return o1 == null ? o2 == null : o1.equals(o2);
}

private static void fatal(String msg) {
 System.out.println(msg);
 System.exit(1);
}

private static void ensure(boolean cond, String msg) {
 if (cond) return;
 fatal(msg);
}

private static Object[] k, v, nk;
private static KEY_TYPE kt[];
private static KEY_TYPE nkt[];
private static RB_TREE_SET topSet;

protected static void testSets(SORTED_SET m, SortedSet t, int n, int level)
{
 long ms;
 boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement;
 boolean rt = false, rm = false;
```

```
  if (level > 4) return;


  /* Now we check that both sets agree on first/last keys. */

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.first();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.first();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): first() divergence at start in
NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  if (! mThrowsNoElement) ensure(t.first().equals(m.first()), "Error (" + level + ", " + seed + "): m and t differ at start
on their first key (" + m.first() + ", " + t.first() +")");

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.last();
  }
  catch (NoSuchElementException
 e) { mThrowsNoElement = true; }
  try {
   t.last();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): last() divergence at start in
NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");


  if (! mThrowsNoElement) ensure(t.last().equals(m.last()), "Error (" + level + ", " + seed + "): m and t differ at start
on their last key (" + m.last() + ", " + t.last() +")");


  /* Now we check that m and t are equal. */
  if (!m.equals(t) || ! t.equals(m)) System.err.println("m: " + m + " t: " + t);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");
```

```
  /* Now we check that m actually holds that data. */
  for(Iterator i=t.iterator(); i.hasNext();) {
   ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry
 after insertion (iterating on t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */
  for(Iterator i=m.iterator(); i.hasNext();) {
   ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
m)");
  }

  /* Now we check that inquiries about random data give the same answer in m and t. For
     m we use the polymorphic method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

   try {
    m.contains(T);
   }
   catch (NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }

   try {
    t.contains(KEY2OBJ(T));
   }
   catch (NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains()
 divergence in NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
   if (!mThrowsNoElement && !mThrowsIllegal) ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)),
"Error (" + level + ", " + seed + "): divergence in keys between t and m (polymorphic method)");
  }

  /* Again, we check that inquiries about random data give the same answer in m and t, but
     for m we use the standard method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;
```

```
  try {
   m.contains(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }

  try {
   t.contains(KEY2OBJ(T));
  }
  catch (NoSuchElementException
 e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  if (!mThrowsNoElement && !mThrowsIllegal) ensure(m.contains(KEY2OBJ(T)) ==  t.contains(KEY2OBJ(T)),
"Error (" + level + ", " + seed + "): divergence between t and m (standard method)");
  }

  /* Now we add and remove random data in m and t, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
  KEY_TYPE T = genKey();

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   rm = m.add(KEY2OBJ(T));
  }
  catch (NoSuchElementException
 e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }

  try {
   rt = t.add(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }


  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  if (!mThrowsNoElement && !mThrowsIllegal) ensure(rm == rt, "Error (" + level + ", " + seed + "): divergence in
```

```
add() between t and m");

  T = genKey();

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   rm = m.remove(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { mThrowsNoElement
 = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }

  try {
   rt = t.remove(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }


  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  if (!mThrowsNoElement && !mThrowsIllegal) ensure(rm == rt, "Error (" + level + ", " + seed + "): divergence in
remove() between t and m");
  }

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal");

  /*
 Now we check that m actually holds the same data. */

  for(Iterator i=t.iterator(); i.hasNext();) {
   ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */

  for(Iterator i=m.iterator(); i.hasNext();) {
   ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
m)");
  }

  /* Now we check that both sets agree on first/last keys. */
```

```
  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.first();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.first();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): first() divergence in
NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement
 + ")");
  if (! mThrowsNoElement) ensure(t.first().equals(m.first()), "Error (" + level + ", " + seed + "): m and t differ on their
first key (" + m.first() + ", " + t.first() +")");

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.last();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.last();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): last() divergence in
NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");

  if (! mThrowsNoElement) ensure(t.last().equals(m.last()), "Error (" + level + ", " + seed + "): m and t differ on their
last key (" + m.last() + ", " + t.last() +")");

  /* Now we check cloning. */

  if (level == 0) {
   ensure(m.equals(((RB_TREE_SET)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
   ensure(((RB_TREE_SET)m).clone().equals(m),
 "Error (" + level + ", " + seed + "): m.clone() does not equal m");
   m = (RB_TREE_SET)((RB_TREE_SET)m).clone();
  }

  /* Now we play with constructors. */
  ensure(m.equals(new RB_TREE_SET((Collection)m)), "Error (" + level + ", " + seed + "): m does not equal new
(Collection m)");
  ensure((new RB_TREE_SET((Collection)m)).equals(m), "Error (" + level + ", " + seed + "): new (Collection
m)does not equal m");
```

```
  ensure(m.equals(new RB_TREE_SET((COLLECTION)m)), "Error (" + level + ", " + seed + "): m does not equal
new (type-specific Collection m)");
  ensure((new RB_TREE_SET((COLLECTION)m)).equals(m), "Error (" + level + ", " + seed + "): new (type-specific
Collection m) does not equal m");
  ensure(m.equals(new RB_TREE_SET((SortedSet)m)), "Error (" + level + ", " + seed + "): m does not equal new
(SortedSet m)");
  ensure((new RB_TREE_SET((SortedSet)m)).equals(m), "Error (" + level + ", " + seed + "): new (SortedSet m) does
not equal m");
  ensure(m.equals(new RB_TREE_SET((SORTED_SET)m)),
 "Error (" + level + ", " + seed + "): m does not equal new (type-specific SortedSet m)");
  ensure((new RB_TREE_SET((SORTED_SET)m)).equals(m), "Error (" + level + ", " + seed + "): new (type-specific
SortedSet m) does not equal m");
  ensure(m.equals(new RB_TREE_SET(m.iterator())), "Error (" + level + ", " + seed + "): m does not equal new
(m.iterator())");
  ensure((new RB_TREE_SET(m.iterator())).equals(m), "Error (" + level + ", " + seed + "): new (m.iterator()) does
not equal m");
  ensure(m.equals(new RB_TREE_SET(m.iterator())), "Error (" + level + ", " + seed + "): m does not equal new
(m.type_specific_iterator())");
  ensure((new RB_TREE_SET(m.iterator())).equals(m), "Error (" + level + ", " + seed + "): new
(m.type_specific_iterator()) does not equal m");

  /* Now we play with conversion to array, wrapping and copying. */
  ensure(m.equals(new RB_TREE_SET(m.TO_KEY_ARRAY())), "Error (" + level + ", " + seed + "): m does not
equal new (toArray(m))");
  ensure((new
 RB_TREE_SET(m.TO_KEY_ARRAY())).equals(m), "Error (" + level + ", " + seed + "): new (toArray(m)) does
not equal m");



  int h = m.hashCode();


  /* Now we save and read m. */

  SORTED_SET m2 = null;

  try {
   java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
   java.io.OutputStream os = new java.io.FileOutputStream(ff);
   java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

   oos.writeObject(m);
   oos.close();

   java.io.InputStream is = new java.io.FileInputStream(ff);
   java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);
```

```
 m2 = (SORTED_SET)ois.readObject();
 ois.close();
 ff.delete();
}
catch(Exception e) {
 e.printStackTrace();
 System.exit(1);
}

ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

/* Now we check that m2 actually holds that data. */

ensure(m2.equals(t), "Error (" + level + ", " + seed +
"): ! m2.equals(t) after save/read");
ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");

/* Now we take out of m everything, and check that it is empty. */

for(Iterator i=t.iterator(); i.hasNext();) m2.remove(i.next());

ensure(m2.isEmpty(), "Error (" + level + ", " + seed + "): m2 is not empty (as it should be)");

/* Now we play with iterators. */

{
 java.util.ListIterator i, j;
 Object J;
 i = (java.util.ListIterator)m.iterator();
 j = new java.util.LinkedList(t).listIterator();

 for(int k = 0; k < 2*n; k++) {
  ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext()");
  ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious()");

  if (r.nextFloat() < .8 && i.hasNext()) {
   ensure(i.next().equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next()");

   if (r.nextFloat() < 0.5) {
    i.remove();
    j.remove();
    t.remove(J);
   }
  }
  else
if (r.nextFloat() < .2 && i.hasPrevious()) {
   ensure(i.previous().equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous()");
```

```
    if (r.nextFloat() < 0.5) {
     i.remove();
     j.remove();
     t.remove(J);
    }
   }

   ensure(i.nextIndex() == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
   ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

  }

 }

 {
  boolean badPrevious = false;
  Object previous = null;
  it.unimi.dsi.fastutil.BidirectionalIterator i;
  java.util.ListIterator j;
  Object I, J;
  KEY_TYPE from = genKey();
  j = new java.util.LinkedList(t).listIterator();
  while(j.hasNext()) {
   Object k = j.next();
   if (((Comparable)k).compareTo(KEY2OBJ(from)) > 0) {
    badPrevious = true;
    j.previous();
    break;
   }
   previous = k;
  }

  i
 = (it.unimi.dsi.fastutil.BidirectionalIterator)m.iterator(from);

  for(int k = 0; k < 2*n; k++) {
   ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting
point " + from + ")");
   ensure(i.hasPrevious() == j.hasPrevious() || badPrevious && (i.hasPrevious() == (previous != null)), "Error (" +
level + ", " + seed + "): divergence in hasPrevious() (iterator with starting point " + from + ")" + badPrevious);

   if (r.nextFloat() < .8 && i.hasNext()) {
    ensure((I = i.next()).equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next() (" + I + ", " + J + ",
iterator with starting point " + from + ")");
    //System.err.println("Done next " + I + " " + J + "  " + badPrevious);

    badPrevious = false;
```

```
    if (r.nextFloat() < 0.5) {
     //System.err.println("Removing in next");
     i.remove();
     j.remove();
     t.remove(J);
    }
   }
   else if (!badPrevious && r.nextFloat()
< .2 && i.hasPrevious()) {
    ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

    if (r.nextFloat() < 0.5) {
     //System.err.println("Removing in prev");
     i.remove();
     j.remove();
     t.remove(J);
    }
   }
  }

 }

 /* Now we check that m actually holds that data. */

 ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
 ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after iteration");

 /* Now we select a pair of keys and create a subset. */

 if (! m.isEmpty()) {
  java.util.ListIterator i;
  Object start = m.first(), end = m.first();
  for(i = (java.util.ListIterator)m.iterator(); i.hasNext() && r.nextFloat() < .3; start = end = i.next());
  for(; i.hasNext() && r.nextFloat() < .95; end = i.next());

  //System.err.println("Checking subSet from
" + start + " to " + end + " (level=" + (level+1) + ")...");
  testSets((SORTED_SET)m.subSet((KEY_CLASS)start, (KEY_CLASS)end), t.subSet(start, end), n, level + 1);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after subSet");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subSet");

  //System.err.println("Checking headSet to " + end + " (level=" + (level+1) + ")...");
  testSets((SORTED_SET)m.headSet((KEY_CLASS)end), t.headSet(end), n, level + 1);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after headSet");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after headSet");
```

```
  //System.err.println("Checking tailSet from " + start + " (level=" + (level+1) + ")...");
  testSets((SORTED_SET)m.tailSet((KEY_CLASS)start), t.tailSet(start), n, level + 1);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after tailSet");
  ensure(t.equals(m),
"Error (" + level + ", " + seed + "): ! t.equals(m) after tailSet");
  }


 }


 private static void runTest(int n) {
  RB_TREE_SET m = new RB_TREE_SET();
  SortedSet t = new java.util.TreeSet();
  topSet = m;
  k = new Object[n];
  nk = new Object[n];
  kt = new KEY_TYPE[n];
  nkt = new KEY_TYPE[n];

  for(int i = 0; i < n; i++) {
#if KEY_CLASS_Object
   k[i] = kt[i] = genKey();
   nk[i] = nkt[i] = genKey();
#else
   k[i] = new KEY_CLASS(kt[i] = genKey());
   nk[i] = new KEY_CLASS(nkt[i] = genKey());
#endif
  }

  /* We add pairs to t. */
  for(int i = 0; i < n;  i++) t.add(k[i]);

  /* We add to m the same data */
  m.addAll(t);

  testSets(m, t, n, 0);

  System.out.println("Test OK");
  return;
 }


 public static void main(String args[]) {
  int n  = Integer.parseInt(args[1]);
  if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));
```

```
  try {
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
   else
 if ("test".equals(args[0])) runTest(n);
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed: " + seed);
  }
 }

#endif


}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/RBTreeSet.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import it.unimi.dsi.fastutil.BigList;
import it.unimi.dsi.fastutil.BigArrays;
import java.util.Collection;
import java.util.List;
import java.util.Random;

/** A class providing static methods and objects that do useful things with type-specific big lists.
*
```

```
* @see java.util.Collections
* @see it.unimi.dsi.fastutil.BigList
 */

public final class BIG_LISTS {

 private BIG_LISTS() {}

 /** Shuffles the specified big list using the specified pseudorandom number generator.
  *
  * @param l the big list to be shuffled.
  * @param random a pseudorandom number generator.
  * @return {@code l}.
  */
 public static KEY_GENERIC BIG_LIST KEY_GENERIC shuffle(final BIG_LIST KEY_GENERIC l, final
Random random) {
  for(long i = l.size64(); i-- != 0;) {
   final long p = (random.nextLong() & 0x7FFFFFFFFFFFFFFFL) % (i + 1);
   final KEY_GENERIC_TYPE t = l.GET_KEY(i);
   l.set(i, l.GET_KEY(p));
   l.set(p, t);
  }
  return l;
 }


 /** An immutable class representing an empty type-specific big list.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific list.
  */

 public static class EmptyBigList KEY_GENERIC extends COLLECTIONS.EmptyCollection KEY_GENERIC
implements BIG_LIST KEY_GENERIC, java.io.Serializable, Cloneable {

  private static final long
 serialVersionUID = -7046029254386353129L;

  protected EmptyBigList() {}

  @Override
  public KEY_GENERIC_TYPE GET_KEY(long i) { throw new IndexOutOfBoundsException(); }

  @Override
  public boolean REMOVE(KEY_TYPE k) { throw new UnsupportedOperationException(); }

  @Override
  public KEY_GENERIC_TYPE REMOVE_KEY(long i) { throw new UnsupportedOperationException(); }
```

```
  @Override
  public void add(final long index, final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException();
}

  @Override
  public KEY_GENERIC_TYPE set(final long index, final KEY_GENERIC_TYPE k) { throw new
UnsupportedOperationException(); }

  @Override
  public long indexOf(KEY_TYPE k) { return -1; }

  @Override
  public long lastIndexOf(KEY_TYPE k) { return -1; }

  @Override
  public boolean addAll(long i, Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
  @Override
  public boolean addAll(COLLECTION c) { throw new UnsupportedOperationException();
 }

  @Override
  public boolean addAll(BIG_LIST c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(long i, COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(long i, BIG_LIST c) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public void add(final long index, final KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException();
}

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public boolean add(final KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
```

```
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS get(long i) { throw new IndexOutOfBoundsException(); }

  /** {@inheritDoc}

 * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS set(final long index, final KEY_GENERIC_CLASS k) { throw new
UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS remove(long k) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public long indexOf(Object k) { return -1; }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public long lastIndexOf(Object k) { return -1; }
#endif

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
  public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator() { return
BIG_LIST_ITERATORS.EMPTY_BIG_LIST_ITERATOR; }

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
  public
 KEY_BIG_LIST_ITERATOR KEY_GENERIC iterator() { return
BIG_LIST_ITERATORS.EMPTY_BIG_LIST_ITERATOR; }

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
  public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(long i) { if (i == 0) return
BIG_LIST_ITERATORS.EMPTY_BIG_LIST_ITERATOR; throw new
IndexOutOfBoundsException(String.valueOf(i)); }
```

```
  @Override
  public BIG_LIST KEY_GENERIC subList(long from, long to) { if (from == 0 && to == 0) return this; throw new
IndexOutOfBoundsException(); }

  @Override
  public void getElements(long from, KEY_TYPE[][] a, long offset, long length) { BigArrays.ensureOffsetLength(a,
offset, length); if (from != 0) throw new IndexOutOfBoundsException(); }

  @Override
  public void removeElements(long from, long to) { throw new UnsupportedOperationException(); }

  @Override
  public void addElements(long index, final KEY_GENERIC_TYPE a[][], long offset, long length) { throw new
UnsupportedOperationException(); }

  @Override
  public void addElements(long
 index, final KEY_GENERIC_TYPE a[][]) { throw new UnsupportedOperationException(); }

  @Override
  public void size(long s)  { throw new UnsupportedOperationException(); }

  @Override
  public long size64() { return 0; }

#if ! KEY_CLASS_Reference
  @Override
  public int compareTo(final BigList<? extends KEY_GENERIC_CLASS> o) {
   if (o == this) return 0;
   return ((BigList<?>)o).isEmpty() ? 0 : -1;
  }
#endif
  @Override
  public Object clone() { return EMPTY_BIG_LIST; }

  @Override
  public int hashCode() { return 1; }

  @Override
  @SuppressWarnings("rawtypes")
  public boolean equals(Object o) { return o instanceof BigList && ((BigList)o).isEmpty(); }

  @Override
  public String toString() { return "[]"; }

  private Object readResolve() { return EMPTY_BIG_LIST; }
 }
```

```
 /** An empty big list (immutable). It is serializable and cloneable.
  */
 SUPPRESS_WARNINGS_KEY_RAWTYPES
 public static final EmptyBigList EMPTY_BIG_LIST = new EmptyBigList();

#if KEYS_REFERENCE
 /**
 Returns an empty big list (immutable). It is serializable and cloneable.
  *
  * <p>This method provides a typesafe access to {@link #EMPTY_BIG_LIST}.
  * @return an empty big list (immutable).
  */
 @SuppressWarnings("unchecked")
 public static KEY_GENERIC BIG_LIST KEY_GENERIC emptyList() {
  return EMPTY_BIG_LIST;
 }
#endif

 /** An immutable class representing a type-specific singleton big list.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific big list.
  */

 public static class Singleton KEY_GENERIC extends ABSTRACT_BIG_LIST KEY_GENERIC implements
java.io.Serializable, Cloneable {

  private static final long serialVersionUID = -7046029254386353129L;

  private final KEY_GENERIC_TYPE element;

  protected Singleton(final KEY_GENERIC_TYPE element) { this.element = element; }

  @Override
  public KEY_GENERIC_TYPE GET_KEY(final long i) { if (i == 0) return element; throw new
IndexOutOfBoundsException(); }

  @Override
  public boolean
 REMOVE(KEY_TYPE k) { throw new UnsupportedOperationException(); }

  @Override
  public KEY_GENERIC_TYPE REMOVE_KEY(final long i) { throw new UnsupportedOperationException(); }

  @Override
  public boolean contains(final KEY_TYPE k) { return KEY_EQUALS(k, element); }

  /* Slightly optimized w.r.t. the one in ABSTRACT_SET. */
```

```
  @Override
  public KEY_TYPE[] TO_KEY_ARRAY() {
   KEY_TYPE a[] = new KEY_TYPE[1];
   a[0] = element;
   return a;
  }

  @Override
  public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator() { return
BIG_LIST_ITERATORS.singleton(element); }

  @Override
  public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(long i) {
   if (i > 1 || i < 0) throw new  IndexOutOfBoundsException();
   KEY_BIG_LIST_ITERATOR KEY_GENERIC l = listIterator();
   if (i == 1) l.NEXT_KEY();
   return l;
  }

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
  public BIG_LIST KEY_GENERIC subList(final long from, final long to) {
   ensureIndex(from);
   ensureIndex(to);
   if (from > to) throw
 new IndexOutOfBoundsException("Start index (" + from + ") is greater than end index (" + to + ")");

   if (from != 0 || to != 1) return EMPTY_BIG_LIST;
   return this;
  }

  @Override
  public boolean addAll(long i, Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

  @Override
  public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

  @Override
  public boolean removeAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean retainAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
```

```
  @Override
  public boolean addAll(BIG_LIST c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(long i, BIG_LIST c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(long i, COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public
 boolean addAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean removeAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean retainAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

#endif

  @Override
  public void clear() { throw new UnsupportedOperationException(); }

  @Override
  public long size64() { return 1; }

  @Override
  public Object clone() { return this; }
 }

 /** Returns a type-specific immutable big list containing only the specified element. The returned big list is
serializable and cloneable.
  *
  * @param element the only element of the returned big list.
  * @return a type-specific immutable big list containing just {@code element}.
  */

 public static KEY_GENERIC BIG_LIST KEY_GENERIC singleton(final KEY_GENERIC_TYPE element) {
return new Singleton KEY_GENERIC_DIAMOND(element); }

#if KEYS_PRIMITIVE

 /** Returns a type-specific immutable
 big list containing only the specified element. The returned big list is serializable and cloneable.
  *
  * @param element the only element of the returned big list.
  * @return a type-specific immutable big list containing just {@code element}.
```

```
 */

 public static KEY_GENERIC BIG_LIST KEY_GENERIC singleton(final Object element) { return new Singleton
KEY_GENERIC_DIAMOND(KEY_OBJ2TYPE(element)); }

#endif


 /** A synchronized wrapper class for big lists. */

 public static class SynchronizedBigList KEY_GENERIC extends COLLECTIONS.SynchronizedCollection
KEY_GENERIC implements BIG_LIST KEY_GENERIC, java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final BIG_LIST KEY_GENERIC list; // Due to the large number of methods that are not in
COLLECTION, this is worth caching.

  protected SynchronizedBigList(final BIG_LIST KEY_GENERIC l, final Object sync) {
   super(l, sync);
   this.list = l;
  }

  protected SynchronizedBigList(final
 BIG_LIST KEY_GENERIC l) {
   super(l);
   this.list = l;
  }

  @Override
  public KEY_GENERIC_TYPE GET_KEY(final long i) { synchronized(sync) { return list.GET_KEY(i); } }

  @Override
  public KEY_GENERIC_TYPE set(final long i, final KEY_GENERIC_TYPE k) { synchronized(sync) { return
list.set(i, k); } }

  @Override
  public void add(final long i, final KEY_GENERIC_TYPE k) { synchronized(sync) { list.add(i, k); } }

  @Override
  public KEY_GENERIC_TYPE REMOVE_KEY(final long i) { synchronized(sync) { return list.REMOVE_KEY(i);
} }

  @Override
  public long indexOf(final KEY_TYPE k) { synchronized(sync) { return list.indexOf(k); } }

  @Override
  public long lastIndexOf(final KEY_TYPE k) { synchronized(sync) { return list.lastIndexOf(k); } }
```

```
  @Override
  public boolean addAll(final long index, final Collection<? extends KEY_GENERIC_CLASS> c) {
synchronized(sync) { return list.addAll(index, c); } }

  @Override
  public void getElements(final long from, final KEY_TYPE a[][], final
 long offset, final long length) { synchronized(sync) { list.getElements(from, a, offset, length); } }

  @Override
  public void removeElements(final long from, final long to) { synchronized(sync) { list.removeElements(from, to); }
}

  @Override
  public void addElements(long index, final KEY_GENERIC_TYPE a[][], long offset, long length) {
synchronized(sync) { list.addElements(index, a, offset, length); } }

  @Override
  public void addElements(long index, final KEY_GENERIC_TYPE a[][]) { synchronized(sync) {
list.addElements(index, a); } }

  /* {@inheritDoc}
   * @deprecated Use {@link #size64()} instead.
   */
  @Deprecated
  @Override
  public void size(final long size) { synchronized(sync) { list.size(size); } }

  @Override
  public long size64() { synchronized(sync) { return list.size64(); } }

  @Override
  public KEY_BIG_LIST_ITERATOR KEY_GENERIC iterator() { return list.listIterator(); }

  @Override
  public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator() { return list.listIterator();
 }

  @Override
  public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(final long i) { return list.listIterator(i); }

  @Override
  public BIG_LIST KEY_GENERIC subList(final long from, final long to) { synchronized(sync) { return
synchronize(list.subList(from, to), sync); } }

  @Override
  public boolean equals(final Object o) { if (o == this) return true; synchronized(sync) { return list.equals(o); } }
```

```
  @Override
  public int hashCode() { synchronized(sync) { return list.hashCode(); } }

#if ! KEY_CLASS_Reference
  @Override
  public int compareTo(final BigList<? extends KEY_GENERIC_CLASS> o) { synchronized(sync) { return
list.compareTo(o); } }
#endif

#if KEYS_PRIMITIVE
  @Override
  public boolean addAll(final long index, final COLLECTION c) { synchronized(sync) { return list.addAll(index, c);
} }

  @Override
  public boolean addAll(final long index, BIG_LIST l) { synchronized(sync) { return list.addAll(index, l); } }

  @Override
  public boolean addAll(BIG_LIST
 l) { synchronized(sync) { return list.addAll(l); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public void add(final long i, KEY_GENERIC_CLASS k) { synchronized(sync) { list.add(i, k); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS get(final long i) { synchronized(sync) { return list.get(i); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS set(final long index, KEY_GENERIC_CLASS k) { synchronized(sync) { return
list.set(index, k); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS remove(final long i) { synchronized(sync) { return list.remove(i); } }

  /** {@inheritDoc}
   * @deprecated
```

```
 Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public long indexOf(final Object o) { synchronized(sync) { return list.indexOf(o); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public long lastIndexOf(final Object o) { synchronized(sync) { return list.lastIndexOf(o); } }
#endif
 }


 /** Returns a synchronized type-specific big list backed by the given type-specific big list.
  *
  * @param l the big list to be wrapped in a synchronized big list.
  * @return a synchronized view of the specified big list.
  * @see java.util.Collections#synchronizedList(List)
  */
 public static KEY_GENERIC BIG_LIST KEY_GENERIC synchronize(final BIG_LIST KEY_GENERIC l) {
return new SynchronizedBigList KEY_GENERIC_DIAMOND(l); }

 /** Returns a synchronized type-specific big list backed by the given type-specific big list, using an assigned object
to synchronize.
  *
  * @param
 l the big list to be wrapped in a synchronized big list.
  * @param sync an object that will be used to synchronize the access to the big list.
  * @return a synchronized view of the specified big list.
  * @see java.util.Collections#synchronizedList(List)
  */

 public static KEY_GENERIC BIG_LIST KEY_GENERIC synchronize(final BIG_LIST KEY_GENERIC l, final
Object sync) { return new SynchronizedBigList KEY_GENERIC_DIAMOND(l, sync); }



 /** An unmodifiable wrapper class for big lists. */

 public static class UnmodifiableBigList KEY_GENERIC extends COLLECTIONS.UnmodifiableCollection
KEY_GENERIC implements BIG_LIST KEY_GENERIC, java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final BIG_LIST KEY_GENERIC list; // Due to the large number of methods that are not in
COLLECTION, this is worth caching.
```

```
  protected UnmodifiableBigList(final BIG_LIST KEY_GENERIC l) {
   super(l);
   this.list = l;
  }

  @Override
  public KEY_GENERIC_TYPE
 GET_KEY(final long i) { return list.GET_KEY(i); }

  @Override
  public KEY_GENERIC_TYPE set(final long i, final KEY_GENERIC_TYPE k) { throw new
UnsupportedOperationException(); }

  @Override
  public void add(final long i, final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException(); }

  @Override
  public KEY_GENERIC_TYPE REMOVE_KEY(final long i) { throw new UnsupportedOperationException(); }

  @Override
  public long indexOf(final KEY_TYPE k) { return list.indexOf(k); }

  @Override
  public long lastIndexOf(final KEY_TYPE k) { return list.lastIndexOf(k); }

  @Override
  public boolean addAll(final long index, final Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

  @Override
  public void getElements(final long from, final KEY_TYPE a[][], final long offset, final long length) {
list.getElements(from, a, offset, length); }

  @Override
  public void removeElements(final long from, final long to) { throw new UnsupportedOperationException();
 }

  @Override
  public void addElements(long index, final KEY_GENERIC_TYPE a[][], long offset, long length) { throw new
UnsupportedOperationException(); }

  @Override
  public void addElements(long index, final KEY_GENERIC_TYPE a[][]) { throw new
UnsupportedOperationException(); }

  /* {@inheritDoc}
   * @deprecated Use {@link #size64()} instead.
   */
```

```
  @Deprecated
  @Override
  public void size(final long size) { list.size(size); }

  @Override
  public long size64() { return list.size64(); }

  @Override
  public KEY_BIG_LIST_ITERATOR KEY_GENERIC iterator() { return listIterator(); }

  @Override
  public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator() { return
BIG_LIST_ITERATORS.unmodifiable(list.listIterator()); }

  @Override
  public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(final long i) { return
BIG_LIST_ITERATORS.unmodifiable(list.listIterator(i)); }

  @Override
  public BIG_LIST KEY_GENERIC subList(final long from, final long to) { return unmodifiable(list.subList(from,
 to)); }

  @Override
  public boolean equals(final Object o) { if (o == this) return true; return list.equals(o); }

  @Override
  public int hashCode() { return list.hashCode(); }

#if ! KEY_CLASS_Reference
  @Override
  public int compareTo(final BigList<? extends KEY_GENERIC_CLASS> o) { return list.compareTo(o); }
#endif

#if KEYS_PRIMITIVE
  @Override
  public boolean addAll(final long index, final COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(final BIG_LIST l) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(final long index, final BIG_LIST l) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS get(final long i) { return list.get(i); }
```

```
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead.
 */
  @Deprecated
  @Override
  public void add(final long i, KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS set(final long index, KEY_GENERIC_CLASS k) { throw new
UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS remove(final long i) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public long indexOf(final Object o) { return list.indexOf(o); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public long lastIndexOf(final Object o) { return list.lastIndexOf(o); }
#endif
 }


 /**
 Returns an unmodifiable type-specific big list backed by the given type-specific big list.
  *
  * @param l the big list to be wrapped in an unmodifiable big list.
  * @return an unmodifiable view of the specified big list.
  * @see java.util.Collections#unmodifiableList(List)
  */
 public static KEY_GENERIC BIG_LIST KEY_GENERIC unmodifiable(final BIG_LIST KEY_GENERIC l) {
return new UnmodifiableBigList KEY_GENERIC_DIAMOND(l); }

 /** A class exposing a list as a big list. */
```

```
 public static class ListBigList KEY_GENERIC extends ABSTRACT_BIG_LIST KEY_GENERIC implements
java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  private final LIST KEY_GENERIC list;

  protected ListBigList(final LIST KEY_GENERIC list) {
   this.list = list;
  }

  private int intIndex(long index) {
   if (index >= Integer.MAX_VALUE) throw new IndexOutOfBoundsException("This big list is restricted to 32-bit
indices");
   return (int)index;
  }

  @Override
  public
 long size64() { return list.size(); }

  @Override
  public void size(final long size) { list.size(intIndex(size)); }

  @Override
  public KEY_BIG_LIST_ITERATOR KEY_GENERIC iterator() { return
BIG_LIST_ITERATORS.asBigListIterator(list.iterator()); }

  @Override
  public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator() { return
BIG_LIST_ITERATORS.asBigListIterator(list.listIterator()); }

  @Override
  public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(final long index) { return
BIG_LIST_ITERATORS.asBigListIterator(list.listIterator(intIndex(index))); }

  @Override
  public boolean addAll(final long index, final Collection<? extends KEY_GENERIC_CLASS> c) { return
list.addAll(intIndex(index), c); }

  @Override
  public BIG_LIST KEY_GENERIC subList(long from, long to) { return new ListBigList
KEY_GENERIC_DIAMOND(list.subList(intIndex(from), intIndex(to))); }

  @Override
  public boolean contains(final KEY_TYPE key) { return list.contains(key); }

  @Override
```

```
  public KEY_TYPE[]
 TO_KEY_ARRAY() { return list.TO_KEY_ARRAY(); }

  @Override
  public void removeElements(final long from, final long to) { list.removeElements(intIndex(from), intIndex(to)); }

#if KEYS_PRIMITIVE
  /* {@inheritDoc}
   * @deprecated Please use {@code toArray()} instead&mdash;this method is redundant and will be removed in the
future.
   */
  @Deprecated
  @Override
  public KEY_TYPE[] TO_KEY_ARRAY(KEY_TYPE[] a) { return list.toArray(a); }

  @Override
  public boolean addAll(long index, COLLECTION KEY_GENERIC c) { return list.addAll(intIndex(index), c); }

  @Override
  public boolean addAll(COLLECTION KEY_GENERIC c) { return list.addAll(c); }

  @Override
  public boolean addAll(long index, BIG_LIST KEY_GENERIC c) { return list.addAll(intIndex(index), c); }

  @Override
  public boolean addAll(BIG_LIST KEY_GENERIC c) { return list.addAll(c); }

  @Override
  public boolean containsAll(COLLECTION KEY_GENERIC c) { return list.containsAll(c); }

  @Override
  public boolean removeAll(COLLECTION
 KEY_GENERIC c) { return list.removeAll(c); }

  @Override
  public boolean retainAll(COLLECTION KEY_GENERIC c) { return list.retainAll(c); }
#endif
  @Override
  public void add(long index, KEY_GENERIC_TYPE key) { list.add(intIndex(index), key); }

  @Override
  public boolean add(KEY_GENERIC_TYPE key) { return list.add(key); }

  @Override
  public KEY_GENERIC_TYPE GET_KEY(long index) { return list.GET_KEY(intIndex(index)); }

  @Override
  public long indexOf(KEY_TYPE k) { return list.indexOf(k); }
```

```
 @Override
 public long lastIndexOf(KEY_TYPE k) { return list.lastIndexOf(k); }

 @Override
 public KEY_GENERIC_TYPE REMOVE_KEY(long index) { return list.REMOVE_KEY(intIndex(index)); }

 @Override
 public KEY_GENERIC_TYPE set(long index, KEY_GENERIC_TYPE k) { return list.set(intIndex(index), k); }

 @Override
 public boolean isEmpty() { return list.isEmpty(); }

 @Override
 public <T> T[] toArray(T[] a) { return list.toArray(a); }

 @Override
 public boolean containsAll(Collection<?>
c) { return list.containsAll(c); }

 @Override
 public boolean addAll(Collection<? extends KEY_GENERIC_CLASS> c) { return list.addAll(c); }

 @Override
 public boolean removeAll(Collection<?> c) { return list.removeAll(c); }

 @Override
 public boolean retainAll(Collection<?> c) { return list.retainAll(c); }

 @Override
 public void clear() { list.clear(); }

 @Override
 public int hashCode() { return list.hashCode(); }
}

/** Returns a big list backed by the specified list.
 *
 * @param list a list.
 * @return a big list backed by the specified list.
 */
 public static KEY_GENERIC BIG_LIST KEY_GENERIC asBigList(final LIST KEY_GENERIC list) { return new
ListBigList KEY_GENERIC_DIAMOND(list); }



#ifdef TEST
```

```
 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif KEY_CLASS_Object
  return Integer.toBinaryString(r.nextInt());
#else
  return
 new java.io.Serializable() {};
#endif
 }


 private static void testLists(KEY_TYPE k, BIG_LIST m, BIG_LIST t, int level) {
  int n = 100;
  int c;

  long ms;
  boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement, mThrowsIndex, tThrowsIndex,
mThrowsUnsupp, tThrowsUnsupp;
  boolean rt = false, rm = false;
  Object Rt = null, Rm = null;

  if (level == 0) return;

  /* Now we check that m and t are equal. */
  if (!m.equals(t) || ! t.equals(m)) System.err.println("m: " + m + " t: " + t);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");

  /* Now we check that m actually holds that data. */
  for(java.util.Iterator i=t.iterator(); i.hasNext();) {
   ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
t)");
  }

  /* Now we check that m actually holds that data, but
 iterating on m. */
  for(java.util.Iterator i=m.listIterator(); i.hasNext();) {
   ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
m)");
  }

  /* Now we check that inquiries about random data give the same answer in m and t. For
     m we use the polymorphic method. */
```

```
  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    m.contains(T);
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }

   try {
    t.contains(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException
 e) { tThrowsIndex = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex) ensure(m.contains(KEY2OBJ(T)) ==
t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in keys between t and m (polymorphic
method) " + m);
  }

  /* Again, we check that inquiries about random data give the same answer in m and t, but
     for m we use the standard
 method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    m.contains(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
```

```
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    t.contains(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
java.util.NoSuchElementException for " + T
 + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): contains() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp)
ensure(m.contains(KEY2OBJ(T)) ==  t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence
between t and m (standard method) " + m);
  }

  /* Now we add and remove random data in m and t, checking that the result is the same. */

  for(int
 i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    rm = m.add(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    rt = t.add(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
```

```
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement
 + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): add() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): add() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in add() between t and m " + m);

   T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    rm = m.remove(KEY2OBJ(T));
   }
   catch
 (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    rt = t.remove(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   if (! KEY_EQUALS(T, k) && mThrowsUnsupp && ! tThrowsUnsupp) mThrowsUnsupp = true; // Stupid bug in
Collections.singleton()

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal,
 "Error (" + level + ", " + seed + "): remove() divergence in IllegalArgumentException for " + T + " (" +
mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): remove() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): remove() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
```

```
  if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in remove() between t and m " + m);
  }

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal " + m);
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal " + m);

  /* Now we add and remove random data in m and t at specific positions,
 checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   int pos = r.nextInt(2);

   try {
    m.add(pos, KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    t.add(pos, KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add()
 at " + pos + " divergence in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " +
tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() at " + pos + " divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): add() at " + pos + " divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): add() at " + pos + " divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);

   T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;
```

```
  pos = r.nextInt(2);

  try {
   Rm = m.remove(pos);
  }
  catch
 (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
  catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

  try {
   Rt = t.remove(pos);
  }
  catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
  catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }


  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() at " + pos + "
divergence in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement
+ ") " + m);
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() at " + pos + " divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal +
 ", " + tThrowsIllegal + ") " + m);
  ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): remove() at " + pos + " divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
  ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): remove() at " + pos + " divergence
in UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
  if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(Rm == Rt || Rm
!= null && Rm.equals(Rt), "Error (" + level + ", " + seed + "): divergence in remove() at " + pos + " between t and
m " + m);
  }

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal " + m);
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal " + m);

  /* Now we add and remove random collections in m and t, checking that the result is the same. */

  for(int i=0;
 i<20*n;  i++) {
  KEY_TYPE T = genKey();

  mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;
```

```
  try {
   rm = m.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
  }
  catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
  catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

  try {
   rt = t.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
  }
  catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
  catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): addAll() divergence
 in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " +
m);
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): addAll() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
  ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): addAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
  ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): addAll() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
  if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in addAll() between t and m " + m);

  T = genKey();

  mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal
 = mThrowsUnsupp = tThrowsUnsupp = false;

  try {
   rm = m.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
  }
  catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
  catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

  try {
   rt = t.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
  }
  catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
  catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }
```

```
   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): removeAll() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal
 == tThrowsIllegal, "Error (" + level + ", " + seed + "): removeAll() divergence in IllegalArgumentException for " +
T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): removeAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): removeAll() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in removeAll() between t and m " + m);
  }

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after set removal " + m);
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after set removal " + m);

  /* Now we add random
 collections at specific positions in m and t, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   int pos = r.nextInt(2);

   try {
    rm = m.addAll(pos, java.util.Collections.singleton(KEY2OBJ(T)));
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    rt = t.addAll(pos, java.util.Collections.singleton(KEY2OBJ(T)));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException
 e) { tThrowsUnsupp = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): addAll() at " + pos + "
divergence in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement
```

```
+ ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): addAll() at " + pos + " divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): addAll() at " + pos + " divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): addAll() at " + pos + " divergence
in UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm
 == rt, "Error (" + level + ", " + seed + "): divergence in addAll() at " + pos + " between t and m " + m);

  }

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after set removal " + m);
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after set removal " + m);

  /* Now we check that m actually holds the same data. */

  for(java.util.Iterator i=t.iterator(); i.hasNext();) {
   ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */

  for(java.util.Iterator i=m.listIterator(); i.hasNext();) {
   ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
m)");
  }

  if (m instanceof Singleton) {
   ensure(m.equals(((Singleton)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
   ensure(((Singleton)m).clone().equals(m),
 "Error (" + level + ", " + seed + "): m.clone() does not equal m");
  }

  int h = m.hashCode();

  /* Now we save and read m. */

  BIG_LIST m2 = null;

  try {
   java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
   java.io.OutputStream os = new java.io.FileOutputStream(ff);
   java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

   oos.writeObject(m);
   oos.close();
```

```
  java.io.InputStream is = new java.io.FileInputStream(ff);
  java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

  m2 = (BIG_LIST)ois.readObject();
  ois.close();
  ff.delete();
 }
 catch(Exception e) {
  e.printStackTrace();
  System.exit(1);
 }

#if ! KEY_CLASS_Reference

 ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

 /* Now we check that m2 actually holds that data. */

 ensure(m2.equals(t), "Error (" + level + ", " + seed + "):
! m2.equals(t) after save/read");
 ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
#endif

 if (! m.isEmpty()) {
  long start = (r.nextLong() & 0x7FFFFFFFFFFFFFFL) % m.size64();
  long end = start + (r.nextLong() & 0x7FFFFFFFFFFFFFFL) % (m.size64() - start);
  //System.err.println("Checking subList from " + start + " to " + end + " (level=" + (level+1) + ")...");
  testLists(k, m.subList(start, end), t.subList(start, end), level - 1);

  ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after subList");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subList");

 }

 return;
}

private static void test() {
 KEY_TYPE k = genKey();
 BIG_LIST m = new Singleton(k);
 BIG_LIST u = BIG_LISTS.unmodifiable(BIG_LISTS.asBigList(LISTS.singleton(KEY2OBJ(k))));
 testLists(k, m, u, 3);
 System.out.println("Test OK");
}

private static long seed = System.currentTimeMillis();
private
```

```
 static java.util.Random r = new java.util.Random(seed);

 private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
 private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

 private static String format(double d) {
  StringBuffer s = new StringBuffer();
  return format.format(d, s, fp).toString();
 }

 private static void fatal(String msg) {
  System.out.println(msg);
  System.exit(1);
 }

 private static void ensure(boolean cond, String msg) {
  if (cond) return;
  fatal(msg);
 }

 /** This method expects as first argument a lower-cased type (e.g., "int"),
  * and as second optional argument a seed. */

 public static void main(String arg[]) throws Exception {
  if (arg.length > 1) r = new java.util.Random(seed = Long.parseLong(arg[1]));

  try {
   test();
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed: " + seed);
  }
 }

#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/BigLists.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2017-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

#if KEYS_PRIMITIVE
import java.util.Objects;
#endif
import java.util.function.Consumer;

/** A type-specific {@link Consumer}; provides methods to consume a primitive type both as object
* and as primitive.
*
* @see Consumer
* @since 8.0.0
*/

@FunctionalInterface
#ifdef JDK_PRIMITIVE_KEY_CONSUMER
public
 interface KEY_CONSUMER KEY_GENERIC extends Consumer<KEY_GENERIC_CLASS>,
JDK_PRIMITIVE_KEY_CONSUMER {
#else
public interface KEY_CONSUMER KEY_GENERIC extends Consumer<KEY_GENERIC_CLASS> {
#endif

#if KEYS_PRIMITIVE
#if !defined JDK_PRIMITIVE_KEY_CONSUMER || KEY_WIDENED
 void accept(KEY_TYPE t);
#endif

#if KEY_WIDENED
 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default void accept(final KEY_TYPE_WIDENED t) {
  accept(KEY_NARROWING(t));
 }
```

```
#endif

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default void accept(final KEY_CLASS t) {
  this.accept(t.KEY_VALUE());
 }

#if !defined JDK_PRIMITIVE_KEY_CONSUMER || KEY_WIDENED
 default KEY_CONSUMER andThen(final KEY_CONSUMER after) {
  Objects.requireNonNull(after);
  return (KEY_TYPE t) -> { accept(t); after.accept(t); };
 }
#endif

#ifdef JDK_PRIMITIVE_KEY_CONSUMER
#if KEY_WIDENED
 /**
 {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
#endif
 @Override
 default KEY_CONSUMER andThen(final JDK_PRIMITIVE_KEY_CONSUMER after) {
  Objects.requireNonNull(after);
  return (KEY_TYPE t) -> { accept(t); after.accept(t); };
 }
#endif

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default Consumer<KEY_GENERIC_CLASS> andThen(final Consumer<? super KEY_GENERIC_CLASS> after)
{
  return Consumer.super.andThen(after);
 }
#endif
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Consumer.drv
No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.Set;

/** A type-specific {@link Set}; provides some additional methods that use polymorphism to avoid (un)boxing.
*
* <p>Additionally, this interface strengthens (again) {@link #iterator()}.
*
* @see Set
*/

public interface SET KEY_GENERIC extends COLLECTION KEY_GENERIC, Set<KEY_GENERIC_CLASS>
 {

 /** Returns a type-specific iterator on the elements of this set.
  *
  * <p>Note that this specification strengthens the one given in {@link java.lang.Iterable#iterator()},
  * which was already strengthened in the corresponding type-specific class,
  * but was weakened by the fact that this interface extends {@link Set}.
  *
  * @return a type-specific iterator on the elements of this set.
  */
 @Override
 KEY_ITERATOR KEY_GENERIC iterator();

#if KEYS_PRIMITIVE
 /** Removes an element from this set.
  *
  * <p>Note that the corresponding method of a type-specific collection is {@code rem()}.
  * This unfortunate situation is caused by the clash
  * with the similarly named index-based method in the {@link java.util.List} interface.
  *
```

```
 * @see java.util.Collection#remove(Object)
 */
boolean remove(KEY_TYPE k);

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@SuppressWarnings("deprecation")
@Deprecated
@Override
default
boolean remove(final Object o) {
 return COLLECTION.super.remove(o);
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@SuppressWarnings("deprecation")
@Deprecated
@Override
default boolean add(final KEY_GENERIC_CLASS o) {
 return COLLECTION.super.add(o);
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@SuppressWarnings("deprecation")
@Deprecated
@Override
default boolean contains(final Object o) {
 return COLLECTION.super.contains(o);
}

/** Removes an element from this set.
 *
 * <p>This method is inherited from the type-specific collection this
 * type-specific set is based on, but it should not used as
 * this interface reinstates {@code remove()} as removal method.
 *
 * @deprecated Please use {@code remove()} instead.
 */
@Deprecated
@Override
default boolean rem(KEY_TYPE k) {
 return remove(k);
}
```

```
#endif
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Set.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import VALUE_PACKAGE.VALUE_COLLECTION;
import VALUE_PACKAGE.VALUE_ABSTRACT_COLLECTION;
import VALUE_PACKAGE.VALUE_ITERATOR;

#if KEYS_PRIMITIVE && VALUES_PRIMITIVE
import it.unimi.dsi.fastutil.objects.ObjectIterator;
#endif
import it.unimi.dsi.fastutil.objects.AbstractObjectSet;

import java.util.Iterator;
import
 java.util.Map;

/** An abstract class providing basic methods for maps implementing a type-specific interface.
*
* <p>Optional operations just throw an {@link
* UnsupportedOperationException}. Generic versions of accessors delegate to
* the corresponding type-specific counterparts following the interface rules
* (they take care of returning {@code null} on a missing key).
*
```

```
* <p>As a further help, this class provides a {@link BasicEntry BasicEntry} inner class
* that implements a type-specific version of {@link java.util.Map.Entry}; it
* is particularly useful for those classes that do not implement their own
* entries (e.g., most immutable maps).
*/

public abstract class ABSTRACT_MAP KEY_VALUE_GENERIC extends ABSTRACT_FUNCTION
KEY_VALUE_GENERIC implements MAP KEY_VALUE_GENERIC, java.io.Serializable {

 private static final long serialVersionUID = -4940583368468432370L;

 protected ABSTRACT_MAP() {}

 @Override
 public boolean containsValue(final VALUE_TYPE v)
 {
  return values().contains(v);
 }

 @Override
 public boolean containsKey(final KEY_TYPE k) {
  final ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> i = ENTRYSET().iterator();
  while(i.hasNext())
   if (i.next().ENTRY_GET_KEY() == k)
    return true;

  return false;
 }

 @Override
 public boolean isEmpty() {
  return size() == 0;
 }

 /** This class provides a basic but complete type-specific entry class for all those maps implementations
  * that do not have entries on their own (e.g., most immutable maps).
  *
  * <p>This class does not implement {@link java.util.Map.Entry#setValue(Object) setValue()}, as the modification
  * would not be reflected in the base map.
  */

 public static class BasicEntry KEY_VALUE_GENERIC implements MAP.Entry KEY_VALUE_GENERIC {
  protected KEY_GENERIC_TYPE key;
  protected VALUE_GENERIC_TYPE value;

  public BasicEntry() {}

  public BasicEntry(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS value) {
```

```
  this.key = KEY_CLASS2TYPE(key);
  this.value
 = VALUE_CLASS2TYPE(value);
  }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

  public BasicEntry(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value) {
   this.key = key;
   this.value = value;
  }

#endif

  @Override
  public KEY_GENERIC_TYPE ENTRY_GET_KEY() {
   return key;
  }

  @Override
  public VALUE_GENERIC_TYPE ENTRY_GET_VALUE() {
   return value;
  }

  @Override
  public VALUE_GENERIC_TYPE setValue(final VALUE_GENERIC_TYPE value) {
   throw new UnsupportedOperationException();
  }

  SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
  @Override
  public boolean equals(final Object o) {
   if (!(o instanceof Map.Entry)) return false;
   if (o instanceof MAP.Entry) {
    final MAP.Entry KEY_VALUE_GENERIC e = (MAP.Entry KEY_VALUE_GENERIC) o;
    return KEY_EQUALS(key, e.ENTRY_GET_KEY()) && VALUE_EQUALS(value, e.ENTRY_GET_VALUE());
   }

   final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
   final Object key = e.getKey();
#if KEYS_PRIMITIVE
   if (key == null || !(key instanceof KEY_CLASS))
 return false;
#endif
   final Object value = e.getValue();
#if VALUES_PRIMITIVE
   if (value == null || !(value instanceof VALUE_CLASS)) return false;
#endif
```

```
  return KEY_EQUALS(this.key, KEY_OBJ2TYPE(key)) && VALUE_EQUALS(this.value,
VALUE_OBJ2TYPE(value));
  }

  @Override
  public int hashCode() {
   return KEY2JAVAHASH(key) ^ VALUE2JAVAHASH(value);
  }

  @Override
  public String toString() {
   return key + "->" + value;
  }
 }

 /** This class provides a basic implementation for an Entry set which forwards some queries to the map.
  */

 public abstract static class BasicEntrySet KEY_VALUE_GENERIC extends AbstractObjectSet<Entry
KEY_VALUE_GENERIC> {
  protected final MAP KEY_VALUE_GENERIC map;

  public BasicEntrySet(final MAP KEY_VALUE_GENERIC map) {
   this.map = map;
  }

  SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
  @Override
  public boolean contains(final Object o) {
   if (!(o instanceof Map.Entry)) return false;

   if (o instanceof MAP.Entry) {
    final MAP.Entry KEY_VALUE_GENERIC
 e = (MAP.Entry KEY_VALUE_GENERIC) o;
    final KEY_GENERIC_TYPE k = e.ENTRY_GET_KEY();
    return map.containsKey(k) && VALUE_EQUALS(map.GET_VALUE(k), e.ENTRY_GET_VALUE());
   }

   final Map.Entry<?, ?> e = (Map.Entry<?, ?>) o;

#if KEYS_PRIMITIVE
   final Object key = e.getKey();
   if (key == null || !(key instanceof KEY_GENERIC_CLASS)) return false;
   final KEY_TYPE k = KEY_OBJ2TYPE(key);
#else
   final Object k = e.getKey();
#endif
   final Object value = e.getValue();
```

```
#if VALUES_PRIMITIVE
   if (value == null || !(value instanceof VALUE_GENERIC_CLASS)) return false;
#endif

   return map.containsKey(k) && VALUE_EQUALS(map.GET_VALUE(k), VALUE_OBJ2TYPE(value));
  }

  SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
  @Override
  public boolean remove(final Object o) {
   if (!(o instanceof Map.Entry)) return false;

   if (o instanceof MAP.Entry) {
    final MAP.Entry KEY_VALUE_GENERIC e = (MAP.Entry KEY_VALUE_GENERIC) o;
    return map.remove(e.ENTRY_GET_KEY(), e.ENTRY_GET_VALUE());
   }

   Map.Entry<?,
 ?> e = (Map.Entry<?, ?>) o;

#if KEYS_PRIMITIVE
   final Object key = e.getKey();
   if (key == null || !(key instanceof KEY_GENERIC_CLASS)) return false;
   final KEY_TYPE k = KEY_OBJ2TYPE(key);
#else
   final Object k = e.getKey();
#endif
#if VALUES_PRIMITIVE
   final Object value = e.getValue();
   if (value == null || !(value instanceof VALUE_GENERIC_CLASS)) return false;
   final VALUE_TYPE v = VALUE_OBJ2TYPE(value);
#else
   final Object v = e.getValue();
#endif

   return map.remove(k, v);
  }

  @Override
  public int size() {
   return map.size();
  }
 }


 /** Returns a type-specific-set view of the keys of this map.
  *
  * <p>The view is backed by the set returned by {@link Map#entrySet()}. Note that
```

```
 * <em>no attempt is made at caching the result of this method</em>, as this would
 * require adding some attributes that lightweight implementations would
 * not need. Subclasses may easily override this policy by calling
 * this method and
caching the result, but implementors are encouraged to
 * write more efficient ad-hoc implementations.
 *
 * @return a set view of the keys of this map; it may be safely cast to a type-specific interface.
 */
@Override
public SET KEY_GENERIC keySet() {
 return new ABSTRACT_SET KEY_GENERIC() {
   @Override
   public boolean contains(final KEY_TYPE k) { return containsKey(k); }
   @Override
   public int size() { return ABSTRACT_MAP.this.size(); }
   @Override
   public void clear() { ABSTRACT_MAP.this.clear(); }
   @Override
   public KEY_ITERATOR KEY_GENERIC iterator() {
    return new KEY_ITERATOR KEY_GENERIC() {
      private final ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> i =
MAPS.fastIterator(ABSTRACT_MAP.this);
      @Override
      public KEY_GENERIC_TYPE NEXT_KEY() { return i.next().ENTRY_GET_KEY(); };
      @Override
      public boolean hasNext() { return i.hasNext(); }
      @Override
      public void remove() { i.remove(); }
     };
   }
  };
}

/**
Returns a type-specific-set view of the values of this map.
 *
 * <p>The view is backed by the set returned by {@link Map#entrySet()}. Note that
 * <em>no attempt is made at caching the result of this method</em>, as this would
 * require adding some attributes that lightweight implementations would
 * not need. Subclasses may easily override this policy by calling
 * this method and caching the result, but implementors are encouraged to
 * write more efficient ad-hoc implementations.
 *
 * @return a set view of the values of this map; it may be safely cast to a type-specific interface.
 */
@Override
public VALUE_COLLECTION VALUE_GENERIC values() {
```

```
 return new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {
   @Override
   public boolean contains(final VALUE_TYPE k) { return containsValue(k); }
   @Override
   public int size() { return ABSTRACT_MAP.this.size(); }
   @Override
   public void clear() { ABSTRACT_MAP.this.clear(); }

   @Override
   public VALUE_ITERATOR
 VALUE_GENERIC iterator() {
    return new VALUE_ITERATOR VALUE_GENERIC() {
      private final ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> i =
MAPS.fastIterator(ABSTRACT_MAP.this);
      @Override
      public VALUE_GENERIC_TYPE NEXT_VALUE() { return i.next().ENTRY_GET_VALUE(); }
      @Override
      public boolean hasNext() { return i.hasNext(); }
     };
   }
  };
 }

 /** {@inheritDoc} */
 @SuppressWarnings({"unchecked","deprecation"})
 @Override
 public void putAll(final Map<? extends KEY_GENERIC_CLASS,? extends VALUE_GENERIC_CLASS> m) {
  if (m instanceof MAP) {
  ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> i = MAPS.fastIterator((MAP KEY_VALUE_GENERIC)
m);

  while (i.hasNext()) {
   final MAP.Entry KEY_VALUE_EXTENDS_GENERIC e = i.next();
   put(e.ENTRY_GET_KEY(), e.ENTRY_GET_VALUE());
  }
 } else {
  int n = m.size();
  final Iterator<? extends Map.Entry<? extends KEY_GENERIC_CLASS,? extends VALUE_GENERIC_CLASS>> i
= m.entrySet().iterator();
  Map.Entry<?
 extends KEY_GENERIC_CLASS,? extends VALUE_GENERIC_CLASS> e;
  while (n-- != 0) {
   e = i.next();
   put(e.getKey(), e.getValue());
  }
 }
}
```

```
/** Returns a hash code for this map.
 *
 * The hash code of a map is computed by summing the hash codes of its entries.
 *
 * @return a hash code for this map.
 */
@Override
public int hashCode() {
 int h = 0, n = size();
 final ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> i = MAPS.fastIterator(this);

 while(n-- != 0) h += i.next().hashCode();
 return h;
}

@Override
public boolean equals(Object o) {
 if (o == this) return true;
 if (! (o instanceof Map)) return false;

 final Map<?,?> m = (Map<?,?>)o;
 if (m.size() != size()) return false;
 return ENTRYSET().containsAll(m.entrySet());
}

@Override
public String toString() {
 final StringBuilder s = new StringBuilder();
 final ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> i = MAPS.fastIterator(this);
 int n = size();
 MAP.Entry KEY_VALUE_GENERIC e;
 boolean first
= true;

 s.append("{");

 while(n-- != 0) {
  if (first) first = false;
  else s.append(", ");

  e = i.next();

#if KEYS_REFERENCE
  if (this == e.getKey()) s.append("(this map)"); else
#endif
   s.append(String.valueOf(e.ENTRY_GET_KEY()));
  s.append("=>");
#if VALUES_REFERENCE
```

```
   if (this == e.getValue()) s.append("(this map)"); else
#endif
    s.append(String.valueOf(e.ENTRY_GET_VALUE()));
  }

  s.append("}");
  return s.toString();
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractMap.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.ListIterator;

/** A type-specific bidirectional iterator that is also a {@link ListIterator}.
*
* <p>This interface merges the methods provided by a {@link ListIterator} and
* a type-specific {@link it.unimi.dsi.fastutil.BidirectionalIterator}. Moreover, it provides
* type-specific
 versions of {@link ListIterator#add(Object) add()}
* and {@link ListIterator#set(Object) set()}.
*
* @see java.util.ListIterator
* @see it.unimi.dsi.fastutil.BidirectionalIterator
*/
```

```
public interface KEY_LIST_ITERATOR KEY_GENERIC extends KEY_BIDI_ITERATOR KEY_GENERIC,
ListIterator<KEY_GENERIC_CLASS> {

 /**
  * Replaces the last element returned by {@link #next} or
  * {@link #previous} with the specified element (optional operation).
  * @param k the element used to replace the last element returned.
  *
  * <p>This default implementation just throws an {@link UnsupportedOperationException}.
  * @see ListIterator#set(Object)
  */

#if KEYS_REFERENCE
 @Override
#endif
 default void set(final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException(); }

 /**
  * Inserts the specified element into the list (optional operation).
  *
  * <p>This default implementation just throws an {@link UnsupportedOperationException}.
  * @param k the element to insert.
  * @see ListIterator#add(Object)

 */

#if KEYS_REFERENCE
 @Override
#endif
 default void add(final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException(); }

 /**
  * Removes from the underlying collection the last element returned
  * by this iterator (optional operation).
  *
  * <p>This default implementation just throws an {@link UnsupportedOperationException}.
  * @see ListIterator#remove()
  */

 @Override
 default void remove() { throw new UnsupportedOperationException(); }



#if KEYS_PRIMITIVE
 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
```

```
 @Override
 default void set(final KEY_CLASS k) { set(k.KEY_VALUE()); }

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default void add(final KEY_CLASS k) { add(k.KEY_VALUE()); }

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default KEY_GENERIC_CLASS
 next() { return KEY_BIDI_ITERATOR.super.next(); }

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default KEY_GENERIC_CLASS previous() { return KEY_BIDI_ITERATOR.super.previous(); }
#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/ListIterator.drv
No license file was found, but licenses were detected in source scan.

8.4.0

- Limited support for atomic big arrays of integers and longs.

8.3.2

- The documentation and the constructor for hash-based containers
 was allowing a load factor of 1. Thanks to Andy Clegg for
 reporting this bug.

- The copy constructor of array-based containers are now more efficient.
 Thanks to Vladimir Krivosheev for proposing this enhancement.

8.3.1

- Fixed old-standing bug in the remove() method of the key set of
 array-based maps. Thanks to Arnaud Floesser for reporting this bug.

- Fixed bug in ensureCapacity() for big lists based on big arrays.
 Thanks to Nate Klein for reporting this bug.

8.3.0

- Add sort and unstableSort for type-specific lists. Thanks to C. Sean Young for
 implementing this feature.

- Add methods in type-specific arrays that dynamically choose algorithm based
 on input. Thanks to C. Sean Young for implementing this feature.

- Fixed very old-standing bug in trim() for
 hash containers: the
 trimming was never happening due to variable shadowing.

- A number of type-specific methods for comparators have been implemented
 by C. Sean Young.

- Big-array methods which can be told apart by dynamic binding are now
 gathered in it.unimi.dsi.fastutil.BigArrays, and can be accessed very
 easily by static imports. The old versions have been deprecated.

- New methods to unwrap iterators into big arrays.

8.2.3

- Array-based lists were not allocating a backing array even if the
 required capacity was greater than the default capacity, violating
 the contract. Thanks to  for reporting this bug.

- FastMultiByteArrayInputStream had since 2014 the wrong slice size
 (1Ki instead of 1Gi). Thanks to Thibault Allançon for his detailed
 reports, which dug out this old one.

8.2.2

- Fixed small bug in new lazy allocation scheme for array-based lists.
 Resizing would throw an exception in certain circumstances.

- Now strategies must accept a supertype of
 the base type. Thanks to
 shevek@github.com for suggesting this change.

- The Maven source jar does not contain tests anymore.

8.2.1

- Added default modularization to OSGi version, too.

8.2.0

- Added default modularization. Thanks to Joshua Popoff for taking care of this issue.

- Implemented lazy allocation of arrays in array-based lists. Thanks to Zheka Kozlov for suggesting this feature.

- Fixed a long-standing issue: the allocation of big lists based on big arrays could not go beyond 2^31 because of a cut-and-paste bug.

- In line with the JDK, the default initial capacity for lists is 10 and backing arrays (and, in generally, arrays) are automatically enlarged by a 50% factor instead of a 100% factor.

- Included dependency-finding scripts by Tobias Meggendorfer.

8.1.1

- More default methods.

- Fixed lack of proper forEach()/fastForEach() method for entry sets of linked maps (the order of iteration would be random). Thanks to Søren Gjesse for reporting this
bug.

- Fixed lack of proper licensing in test sources.

- A new NO_SMALL_TYPES makefile variable makes it possible to compile a version of fastutil for ints, longs and doubles only. Thanks to Tobias Meggendorfer for implementing this feature.

8.1.0

- WARNING: backward-incompatible name change for a few "compute" methods.

- Implemented efficient new map methods and new iterable methods in hash-based maps.

- A number of minor glitches has been fixed.

- FastIterable now has a fastForEach() method.

- Fixed ancient bug in the remove() method of entry sets of tree-based maps.

- Array-based maps have a working key set and value collection.

8.0.0

- First release for Java 8 only, with implementation of new default methods for iterators and maps. Thanks to Tobias Meggendorfer and Salomon Sickert from the Technical University of Munich for help in the implementation. Structure-specific, more efficient code will be added in the near future. All wrapper (synchronized, etc.) have been updated.

- Abstract type-specific comparators are deprecated: their only abstract method has been pulled up as a default method of the type-specific interface, which makes it possible to use lambda expressions to define type-specific comparators.

- The default return value setter/getter are now optional for functions. In this way, functions can now be defined using lambda expressions (and they implement java.util.function.Function).

- New static methods in type-specific Maps and SortedMaps make it possible to use easily fast iterators, even within for loops. Thanks to Tobias Meggendorfer and Salomon Sickert from the Technical University of Munich for implementing this feature.

- All hash-based structure now keep track of the size of their backing array at creation time, and will never rehash below that size. Thanks to Patrick Julien for suggesting this feature.

- Every deprecated method marked to be removed, or replaced by another method, has been eliminated.

- Many abstract classes are now obsolete as their abstract methods have become default methods.

- The type hierarchy of big-list iterators has been fixed.

7.2.0

- Major code cleanup. Several unnecessary method implementations have been deleted. Several missing methods (e.g., equals()/hashCode() in wrappers) have been implemented. All methods should now have a reasonable Javadoc comment.

- Clarified the rem()/remove() conundrum: type-specific collections should use and override rem(), but type-specific sets should use and override

remove().

- Fixed circular implementation of containsKey() in type-specific abstract collections.

- Pulled up deprecation at the highest possible (interface) level.

- Collection.*All() methods do not assume anymore that they can rely on the argument size().

7.1.1

- Fixed decade-old efficiency bug: implementation of RandomAccess was hidden from synchronized and unmodifiable wrappers. Thanks to Peter Burka for reporting
this bug.

- The defaultReturnValue() getter in unmodifiable maps was throwing an UnsupportedOperationException instead of delegating the call to the underlying map. Thanks to Alex Fiennes for reporting this bug.

7.1.0

- Fixed decade-old efficiency bug. Due to a name clash between lists and sets, the type-specific deletion method of a type-specific collection is rem(), not remove(). The latter is reinstated in sets (but not, for example, in lists) by the type-specific abstract set classes. Nonetheless, implementors of subclasses must override rem(), not remove(), as methods such as the type-specific version of removeAll() invoke necessarily invoke rem() rather than remove(). Up to this version, all concrete set implementations were overriding remove(), instead, causing inefficiencies in the inherited methods. Thanks to Christian Habermehl for reporting this bug.

- Fixed a bug introduced with the removal of old-style gcc assertions: all load methods in
BinIO that did not specify the number of elements to read were computing the number of items in the loaded file incorrectly, causing an EOFException (except for booleans and bytes).

7.0.13

- Fixed inheritance problem that would surface as key sets of maps not implementing remove(). Thanks to Luke Nezda for reporting this bug.

7.0.12

- Collection.isEmpty() was checking for iterator().hasNext() instead of the opposite. Thanks to Olaf Krische for reporting this bug.

- Fixed lack of test for null/wrong class when testing entries.

7.0.11

- Several small glitches that were making fastutil's classes behave differently from those of java.util have been fixed. Thanks to Balázs Attila-Mihály or reporting these bug (obtained by massive testing using Guava's battery of unit tests).

7.0.10

- The infinite-loop bug was affecting trim(int) besides trim(). Thanks to Igor Kabiljo for reporting this bug.

- With the help of Erich Schubert, all methods with a type-specific, more
efficient counterpart have been deprecated.

7.0.9

- A subtle infinite-loop bug in hash-based structures (happening with load factor 1 and tiny structures) has been fixed. Thanks to Tuomas Välimäki and Jarkko Mönkkönen for reporting this bug.

- Now tree-based map have an addTo() method analogous to that of hash-based maps. Thanks to Almog Gavra for implementing the method.

7.0.8

- Non-indirect priority queues are now serializable.

- Fixed implementation of structures based on a custom hash: keys strategy-equal to zero zero would not be managed correctly. Thanks to Shawn Cao for reporting this bug.

- Natural/opposite/abstract comparators are now serializable.

7.0.7

- Now we check whether ranges of parallel sorting algorithms are too small *before* creating the thread pool.

- Merged Erich Schubert's fix for Object{AVL,RB}TreeSet.get().

7.0.6

- Faster priority queues: better variable caching, deleted
 a spurious check, tests for parameters turned into assertions.

-
 New collection-based constructors for heap-based priority queues.

- Reviewed ObjectArrays.newArray() so that there is a fast track
 for reallocation of arrays of type Object[].

7.0.4

- Fixed old-standing bug: iterators in linked maps would
 return bogus data on entrySet().next()/entrySet().previous()
 when no element is available instead of throwing an exception.

7.0.3

- Fixed wrong generation of custom-hash classes with primitive
 keys. Thanks to Michael Henke for reporting this bug.

7.0.2

- Now we shutdown() correctly ForkJoinPool's.

- Constants limiting parallelism and recursion have been tuned.

- New implementations of indirect [parallel] quicksort (in ascending order
 only).

- New stabilization method for post-processing of non-stable indirect
 sorts.

7.0.1

- Now generated sources are formatted using the Eclipse command-line
 facility.

7.0.0

- Now we need Java 7.

- New parallel versions of radix sort and quicksort. The sequential
 implementations have
 been further improved.

- Restored the previous constants in mixing functions.

6.6.4

- Hopefully better mixing functions created by a genetic algorithm.

- Fixed a bug in floating-point hash-based containers: -0.0 and +0.0 were both converted to +0.0. Thanks to Dawid Weiss for reporting this bug.

6.6.3

- Fixed subtle wrap-around bug in removal from iterator. Thanks to Eugene Yakavets for reporting this bug.

6.6.2

- We now reduce backing arrays of hash-based classes when they are filled below one fourth of the load factor. The reduction is not performed when deleting from an iterator, as it would make iteration impossible.

- Significant simplification of Iterator.remove()'s implementations for hash-based data structures.

6.6.1

- Fixed missed implementation: setValue() was not implemented for fast iterators in hash-based maps.

6.6.0

- Major (transparent) rewrite of all hash-based classes inspired by the Goldman-Sachs collections. We no longer allocate a byte array to store
the status of each slot: a null (or zero) key denotes an empty slot. The null key is handled separately. The reduction in memory accesses makes the cost of the additional logic negligible, and brings in significant performance improvements. The code is actually simplified, as all loops become a search for a nonzero element.

- Partial (one-step) unrolling of all lookup loops, following the strategy used in Koloboke.

- Fixed an old bug: entrySet().remove(Entry) would remove entries checking the value of the key, only.

- Fixed a bug in the iterator over hash big sets.

- OSGI metadata, thanks to Benson Margulies.

6.5.17

- Now TextIO methods trim strings before parsing numbers. This avoids obnoxious exceptions when numbers are followed by whitespace.

6.5.16

- Improved speed of FastMultiByteArrayInputStream, and removed support for mark()/reset().

- Deprecated array fill() methods in favour of java.util's.

6.5.15

- De-deprecated quicksort methods for primitive-type arrays. It turned out that Java's Arrays.sort() switches to mergesort on large, semi-sorted arrays. Moreover, in Java 7 the support array is allocated of the same size of the argument array, not of the sorted fragment. This performance bug was entirely killing the performance of Transform.transposeOffline() and other methods. Until that bug is fixed, we will have to rely on our quicksort method (which is a pity, because Java's sort is, for the rest, so beautifully engineered).

6.5.14

- Equality in type-specific hash-based data structures with float or double keys is now checked by converting to int/long bits using the conversion method of the appropriate class. Previously, using NaNs as keys would have led to misbehaviour. Thanks to Davide Savazzi for reporting this bug.

6.5.13

- Fixed a very unlikely corner case that might have led to reduction in size of an array instead of a growth. Thanks to Ernst Reissner for reporting this bug.

- InspectableFileCachedInputStream no longer performs a call to RandomAccessFile.position() when the end of file has been reached and the file is entirely held in memory.

- All front-coded lists now implement java.util.RandomAccess.

6.5.12

- Removed some useless wrapper creation in a few methods of tree-based map

classes.

- Fixed pathological maxFill computation for very small-sized big open hash sets.

6.5.11

- A very old and subtle performance bug in hash-based data structures has been fixed. Backing arrays were allocated using the number of expected elements divided by the load factor. However, since the test for rehashing was fired by equality with the table size multiplied by the load factor, if the expected number of elements multiplied by the load factor was an integer a useless rehash would happen for the very last added element. The only effect was an useless increase in object creation.

6.5.10

- Now iterators in object set constructors are of type Iterator, and not
anymore ObjectIterator. The kind of allowed iterators has been rationalised and made uniform through all classes implementing Set.

6.5.9

- New methods to get a type-specific Iterable from binary or text files.

6.5.8

- Fixed stupid bug in creation of array-based FIFO queues.

6.5.7

- Fixed a very subtle bug in hash-based data structures: addAll() to a newly created structure could require a very long time due to correlation between the positions in structures with different table sizes.

6.5.6

- equals() method between arrays have been deprecated in favour of the java.util.Arrays version, which is intrinsified in recent JVMs.

- InspectableFileCachedInputStream.reopen() makes it possible to read again from the start an instance on which close() was invoked.

6.5.5

- The abstract implementation of equals() between (big) lists now uses type-specific access methods (as the compareTo() method was already doing) to avoid massive boxing/unboxing. Thanks to Adrien Grand for suggesting this improvement.

- FIFO array-based queues are now serializable.

6.5.4

- Further fixes related to NaNs in sorting.

- Fixed very old bug in FastByteArrayOutputStream.write(int). Thanks to Massimo Santini for reporting this bug.

- We now use Arrays.MAX_ARRAY_SIZE, which is equal to Integer.MAX_VALUE minus 8, to bound all array allocations. Previously, it might happen that grow() and other array-related functions could try to allocate an array of size Integer.MAX_VALUE, which is technically correct from the JLS, but will not work on most JVMs. The maximum length we use now is the same value as that used by java.util.ArrayList. Thanks to William Harvey for suggesting this change.

6.5.3

- Corrected erroneous introduction of compare() methods on integral classes (they appeared in Java 7).

6.5.2

- A few changes were necessary to make fastutil behave as Java on NaNs when sorting. Double.compareTo() and Float.compareTo() treat Double.NaN as greater than Double.POSITIVE_INFINITY, and fastutil was not doing it. As part of the change, now all comparisons between primitive types are performed using the compare() method of the wrapper class (microbenchmarks confirmed that there is no speed penalty for that, probably due to inlining or even intrinsification). Thanks to Adam Klein for reporting this bug.

- All quickSort() implementations that do not involve a comparator are now deprecated, as there are equivalent/better versions in java.util.Arrays.

6.5.0 -> 6.5.1

- Now FastBuffered{Input/Output}Stream has a constructor with an explicitly given buffer.

- Abandoned golden-ratio based expansion of arrays and lists in favour of a (more standard) doubling approach.

- Array-based FIFO queues now reduce their capacity automatically by halving when the size becomes one fourth of the length.

- The add() method for open hash maps has been deprecated and replaced by addTo(), as the name choice proved
to be a recipe for disaster.

- New InspectableFileCachedInputStream for caching easily large byte streams partially on file and partially in memory.

- The front() method for semi-indirect heaps took no comparator, but was used in queues in which you could support a comparator. There is now a further version accepting a comparator.

- Serial Version UIDs are now private.

6.4.6 -> 6.5.0

- Fixed type of array hash strategies.

- Fixed use of equals() instead of compareTo() in SemiIndirectHeaps.front(). Thanks to Matthew Hatem for reporting this bug.

- Now we generate custom hash maps for primite types, too (as we were already doing for sets).

6.4.5 -> 6.4.6

- In array-based priority queues changed() would not invalidate the cached index of the smallest element.

6.4.4 -> 6.4.5

- In some very rare circumstances, enumeration of hash sets or maps combined with massive element removal (using the iterator remove() method) could have led to inconsistent enumeration (duplicates
and
missing elements). Thanks to Hamish Morgan for reporting this bug.

6.4.3 -> 6.4.4

- Array-based maps were not implementing correctly entrySet().contains(), and as a consequence equals() between such maps was broken. Thanks to

Benson Margulies for reporting this bug.

6.4.2 -> 6.4.3

- Now array-based priority queue cache their first element. Moreover, they implement the correct type-specific interface.

6.4.1 -> 6.4.2

- Now we have indirect lexicographical radix sort on pairs of arrays, mainly used to compute quickly Kendall's tau.

- New reverse method for arrays (useful for radix descending sorts).

- Radix sort (one or two arrays) for big arrays.

- Now radix sort uses correctly (minimally) sized support arrays when sorting subarrays.

6.4 -> 6.4.1

- Now we have a separate directory, settable in the makefile, to generate sources. This makes Maven integration easier.

- The store methods in TextIO for big arrays were broken.

- Now big-array lists implement the Stack interface.

- Fixed subtle bug in rehash() methods of big hash sets.

6.3 -> 6.4

- WARNING: Indirect queues must obviously have a way to determine whether an index is in the queue. It was an oversight in the interface design that a contains() method was not present. We wook the risk of adding it now. At the same time, we modified remove() so that now returns a boolean specifying whether the index to be removed was actually in the queue, as this is more in line with the Java Collections Framework.

- Removed unused double-priority queue related classes.

- Now array-based sets and maps have a constructor based on java.util.Collection and java.util.Map (as for the other kind of sets and maps).

- New doubly linked implementation for linked hash maps and sets. It uses twice the space for pointers, but mixes well with linear probing, so we

have again constant-time true deletions. Moreover, iterators can be
started from any key in constant time (albeit
the first access to the
index of the list iterator will require a linear scan, unless the
iterator started from the first or the last key). Additional methods
such as getAndMoveToFirst() make the creation of LRU caches very easy.
Thanks to Brien Colwell for donating the code.

- Now object-based array FIFO queues provide deque methods. Moreover,
they clean up the backing array after returning an object or when
performing a clear().

- New get() method in set implementations makes it possible to recover
the actual object int the collection that is equal to the query key.

- A number of bugs were found and fixed by Christian Falz (thanks!). In
all binary search code the "to" parameter was *inclusive*, but the
documentation said *exclusive*, with obvious problems. Hash map
iterators could return under some very subtle and almost irreproducible
circumstances a previously deleted slot. Deleted hash map entries would
return spurious null values.

6.2.2 -> 6.3

- We now
have radix sort. It's much faster than quicksort, but it can
only sort keys in their natural order. There are multiple-array
and indirect (and possibly stable) versions available.

- There are now custom hash sets also for type-specific keys. This makes
it possible to use hash sets to index data indirectly (e.g., using
integer or long just as indices).

- Shuffling static methods for all kinds of (big) list and arrays.

6.2.1 -> 6.2.2

- A new add() method makes the usage of maps as counters easier
and faster.

6.2.0 -> 6.2.1

- A very stupid bug was causing twice the rehashing that was
necessary. Now insertions in hash-based classes are significantly faster.

6.1.0 -> 6.2.0

- A better structure of the scan loop for hash tables borrowed

from HPPC (http://labs.carrotsearch.com/hppc.html) gives some
speed improvement to hash-based classes.

6.0.0 -> 6.1.0

- Hash-based classes have been rewritten using linear probing and
a good hash (MurmurHash3). The old classes can
be still generated
using the target oldsources.

- Bizarre queues (double- and sesqui-indirect) have been removed
from the standard jar, but they can be still generated using the
target oldsources.

5.1.5 -> 6.0.0

- WARNING: the jar file is now fastutil.jar (not fastutil5.jar), again.

- WARNING: now fastutil requires Java 6+.

- fastutil is now released under the Apache License 2.0.

- New framework for big arrays, represented as arrays-of-arrays.
BigArrays and the type-specific counterparts provide static
methods of all kinds.

- New Size64 interface for classes implementing big collections.

- New framework for big lists--lists with longs as indices. The only
present implementation uses big arrays, but, for instance,
Sux4J's succinct lists will be retrofitted to LongBigList
(presently they implement LongBigList from dsiutils, which will
be deprecated).

- List.iterator() now returns a ListIterator. There is no real reason
not to do this, and the API change is
handled from an implementation
viewpoint in AbstractList, so nodoby should really notice.

- New Collections.asCollection(Iterable) method to expose iterables as
collections (missing methods are computed using the iterator). This was
also the occasion to streamline type-specific abstract collections,
which now inherit from java.util.AbstractCollection, so we support
contains, clear, etc. methods as long as there is an iterator.

- Fixed bugged array-based constructors of ArrayMap and ArraySet.

- Fixed bugged put/remove methods in abstract functions. Thanks to

Katja Filippova for reporting this bug.

- New front-coded lists use big arrays, so they can store much more (in fact, unlimited) data. Unfortunately, they are no longer serialisation-compatible with previous versions.

- New MeasurableStream interface that is implemented by MeasurableInputStream and by a new, analogous MeasurableOutputStream.

- Better FastBufferedOutputStream and FastByteArrayOutputStream that are measurable and positionable.

- Now all clone() methods override covariantly the defult return type (Object).

5.1.4 -> 5.1.5

- ArraySet was implementing isEmpty() with inverted logic (thanks to Marko Srdanovic for reporting this bug).

- New constructor for FastMultiByteArrayInputStream: it takes a MeasurableInputStream and uses length() to determine the number of bytes to load into memory.

5.1.3 -> 5.1.4

- The implementation of RepositionableStream in FastByteArrayOutputStream was fraught with a horrendous bug (thanks to Claudio Corsi for reporting), in spite of extensive unit tests.

5.1.2 -> 5.1.3

- A bug existing since the first release was preventing tables larger than 2^30 bits to work (the computation of the next bucket to look at would cause an integer overflow).

- FastByteArrayOutputStream now implements RepositionableStream.

- Type-specific versions of Iterable.

- Some methods (e.g., iterator() and values()) are now explicitly re-strengthened
wherever necessary to avoid complaints about ambiguous method invocations by some compilers.

- The introduction of functions added several bugs to the empty/singleton map classes. Inheriting from the respective function counterparts left several methods underspecified (equals(), etc.). This has been

(hopefully) fixed.

5.1.1 -> 5.1.2

- FastBufferedInputStream now supportw length() by FileChannel-fetching on FileInputStream instances (it already used to support position() by the same mechanism).

5.1.0 -> 5.1.1

- Byte-array MG4J I/O classes have been moved here.

5.0.9 -> 5.1.0

- Fixed documentation for custom/noncustom maps (it was exchanged).

- New type-specify entrySet() methods that avoid complicated casting to get a type-specific entryset. Moreover, now entrySet() can return an object implementing Fast(Sorted)EntrySet to indicate that a fastIterator() method is available. Fast iterators can return always the same Entry object, suitably mutated. We thank Daniel
Ramage for suggesting this feature.

- Several hundreds of new classes generated by the new Function interface, which represent mappings for which the entry set is not enumerable (e.g., hashes). Functions have their usual share of satellite objects (wrappers, etc.). There are no implementations--the main purpose of the new interfaces is to make Sux4J (http://sux.dsi.unimi.it/) more object-oriented.

5.0.8 -> 5.0.9

- Slightly reduced overhead for bound checks in heap-based queues.

- BinIO was loading byte arrays one byte at a time. Now some conditionally compiled code uses bulk-read methods instead. Moreover, horrible kluges to work around Java bug #6478546 have been included.

5.0.7 -> 5.0.8

- Faster array maps and sets: System.arraycopy() is very slow on small arrays (due to inherent costs of calling native code) and reflection-based array creation is a disaster. Now we use object arrays and loops.

- New clone() methods for array-based structures and custom serialisation.

- FastBuffered*Stream has been simplified and streamlined. No more block alignment.

5.0.6 -> 5.0.7

- Better algorithm for front() in heaps.

- New comprehensive collection of array-based maps and sets. The motivation behind such structures is the need for quick, low-footprint data structures for *very* small sets (say, less than 10 elements). For instance, in MG4J we were using sparse reference-based hash tables, but it turned out that System.identityHashCode() is *deadly* slow and scanning linearly an array searching for the desired element is significantly faster.

5.0.5 -> 5.0.6

- Due to erratic and unpredictable behaviour of InputStream.skip(), which does not correspond to its specification and Sun refuses to fix (see bug 6222822; don't be fooled by the “closed, fixed” label), FastBufferedInputStream now peeks at the underlying stream and if it is System.in it uses repeated reads. Moreover, it will use alternatively reads and skips to guarantee
that the number of skipped bytes will be
smaller than requested only if end of file has been reached.

- The insertion and key retrieval methods of hash-based structures are now protected and final.

- New front() method for indirect queues. It retrieves quickly the indices associated to elements equal to the top.

- First JUnit tests.

5.0.4 -> 5.0.5

- Fixed possible overflow in FastBufferedInputStream.available().

- Indirect heaps have faster checks for elements belonging or not to the queue. In particular, we just rely on array access for detecting indices out of bounds. Profiling with LaMa4J showed that in some circumstances checking explicitly the indices were within bounds was taking more time that the actual heap inner workings.

- Fixed obnoxious bug dating to the first fastutil implementation. The macro KEY_EQUALS_HASH(x,h,y), which checks for equality between x and y given that the hash of x is h, was evaluating hashCode() on y without guarantee that

y was non-null. As a result, adding a null to a mapped
followed by the insertion of an element with hash code 0 would have
thrown a NullPointerException. The bug went unobserved for years because
no one use nulls as keys, and was actually detected by a bug in BUbiNG's
code (which was in turn mistakenly inserting nulls in a set).

5.0.3 -> 5.0.4

- Fixed missing declaration of generic type for HASH_STRATEGY.

- A new abstract class, MeasurableInputStream, is used for streams
whose length and current position are always known. This actually
was needed for BUbiNG development.

- New readLine() family of method for reading "lines" directly
from a FastBufferedInputStream.

- In FastBufferedInputStream, reset() has been deprecated in favour
of flush().

- Array-based lists of objects now reallocate the backing array
using reflection *only* if they were created by wrapping. This
won't change the previous behaviour, but at the price of a boolean
per list we have unbelievably
faster array reallocation.

- New explicit fast load factors in Hash.

5.0.2 -> 5.0.3

- Bizarrily, java.util.List re-specifies iterator(), even if it extends
Collection. As a result, we need to re-strengthen it in type-specific lists.

- Fixed new horrible bug introduced by adding Booleans to BinIO and TextIO.
Problem is, I didn't know #assert is cumulative.

5.0.1 -> 5.0.2

- Fixed bug in sorted maps key sets and values that would cause a
stack overflow when calling size() and a few other methods.

- Fixed lack of booleans in BinIO and TextIO.

- BinIO now checks for too large files.

5.0 -> 5.0.1

- In BinIO, it was assumed that .SIZE would give the size of

primitive types in *bytes*. Bad mistake.

4.4.3 -> 5.0

- Java 5 only!

- Support for generics. This led to a number of backward-incompatible changes:
  * toArray(Object[]) does not accept any longer null as an argument;
  * singletons for empty collections (sets, lists, ecc.) are type-specific;
  * iterators
 on sorted collections are bidirectional *by specification*;
  * the new, covariantly stronger methods defined in all interfaces (e.g.,
    iterator() returning a type-specific iterator) are now the default,
    and in the abstract classes the old methods (e.g., objectIterator())
    now just delegate to the standard method, which is the contrary
    of what was happening before: you'll have to turn all methods
    such as objectIterator() in iterator(), etc.
  * all deprecated methods have been dropped.

- Array growth functions now will return the correct empty array for
 object arrays (it used to return ObjectArrays.EMPTY_ARRAY).

- Strategies are generic and no longer required to accept REMOVED.

- Stale references could hang around in the nodePath array for
 Red-Black trees and map; this has been fixed.

- The difference in semantics with the standard toArray(Object[])
 specification, which has always been in place, is now exhaustively
 explained.

- Major code cleanup
 (mostly code deletion) due to passing fastutil
 into Eclipse to check unused code, etc.

4.4.2 -> 4.4.3

- Important bug fix in FastBufferedInputStream.

4.4.1 -> 4.4.2

- New reset() method to invalidate the buffer of a FastBufferedInputStream,
 making it possible to read safely files written by other processes
 (given, of course, that you are synchronising the accesses).

4.4.0 -> 4.4.1

- New parallel-array constructor for all maps. Very useful for

static final map initialisation.

- Following considerations in Jakarta Commons I/O, the standard
buffer size has be lowered to 8Ki.

- Some arguments were declared as DataInputStream instead of
DataInput.

- New methods for reading/writing objects from/to streams.

4.3.2 -> 4.4

- New static containers for reading and writing easily text and binary
data streams. They load/save arrays, iterators etc. to buffered readers
or streams.

- Moved here fast input/output buffered classes from MG4J. This makes
fastutil self-contained.

-
The trivial implementation of the type-specific iterator was missing
from AbstractList.drv (surprisingly, not from the subclass implementation!).

- The sublist implementation in AbstractList.drv is now protected and static.
The attributes are protected, too.

- Now we compare booleans (false<true). As a result, also lists of
booleans do get lexicographical comparability.

- add(k) in AbstractList.drv now calls add(size(), k).

- Fixed error messages for out-of-bound indices in lists.

4.3.1 -> 4.3.2

- Fixed small innocuous bug: a code fragment related to non-linked
hash table was generated for linked hash tables, too, do to a
case type in a preprocessor directive. The code fragment, however,
had no effect.

- Fixed memory leak in OpenHashMap: the remove() method was not clearing
the key (whereas OpenHashSet was).

4.3 -> 4.3.1

- New fully indirect heap-based double priority queues.

- Fixed docs for queues: in 4.3, we were claiming that greater elements

are dequeued first, while the opposite happens.

4.2 -> 4.3

- New full-fledged set of unmodifiable structures *and* iterators.

- Removed about a dozen spurious final method modifiers.

- Made rehash() protected, so that everybody can play with different rehashing strategies.

- trim() in array lists wasn't doing the right thing, because trim(int) wasn't doing it in the first place. Now if n is smaller than the size of the list, we trim at the list size (previously we were doing nothing).

- Analogously, trim() in hash-table-based structures was fixed so that trimming a table below its size will result in rehashing to the minimum possible size.

4.1 -> 4.2

- Improved array methods: now all methods on objects (e.g., grow()) return an array of the same type of the array that was passed to them, similarly to toArray() in collections.

- Fixed missing macro substitution for empty iterator methods. In any case, they were already deprecated.

4.0 -> 4.1

- New classes for custom hashing methods (mainly thought for arrays). Correspondingly, methods for arrays have been implemented in the static containers.

- BasicEntry now throws an UnsupportedOperationException on calls to setValue(). If you ever used that method, you got wierd results, as it does not update the underlying map. The method is now implemented correctly in open hash maps, in which previously did not correctly update the underying map.

- Reimplemented copy of an entire array using clone().

- Fixed a bug in clear() for indirect heaps (the inversion array was not being cleared).

- Indirect priority queue interfaces now feature an optional allChanged() method that can be used to force a complete heap rebuild. It is implemented by

all current array-based and head-based concrete classes.

3.1 -> 4.0

- IMPORTANT: The optimized methods that a type-specific must provide now include an addElements() method that quickly adds an array of elements. As usual, the method is fully
implemented by the type-specific abstract lists.

- IMPORTANT: The abstract generic version of get(), put() and remove() for maps with non-object keys or values now always return null to denote a missing key. They used to return an object-wrapped default return value.

- Completely new and comprehensive implementation of priority queues, both direct and indirect. Implementations are by heaps and by flat arrays. There are also static containers with all relevant heap methods, for people wanting to do their own thing.

- New static containers for comparators.

- All singletons, empty sets and snychronized wrappers are public so you can inherit from them.

- Abstract maps now provide keySet() and values() based on entrySet().

- New abstract classes for sorted sets and maps with delegators to type-specific methods.

- New public methods in Arrays and in type-specific Arrays classes for checking ranges.

- New static methods for type-specific arrays that allow to grow,
enlarge
and trim them with ease.

- Clarified abstract implementation of default return values, and implemented clarified specification. Just a couple of method in hash maps were not already compliant.

- The pour() method now returns a list. The previous version was returning a linked hash set, which was rather nonsensical anyway, since an iterator build on the returned set could have been different from the original iterator. You can always pour an iterator into a set by providing the set explicitly.

- An exception-throwing implementation of some methods in AbstractSet was missing. Same for AbstractCollection, AbstractMap and AbstractList.

- New basic inner entry class for abstract maps, which makes it easier to write

entrySet() methods for classes that do not have their own entries.

- Added missing get(Object) method in AbstractMap (just delegates to
the type-specific version).

- For lazy people, now containsKey() and containsValue() in AbstractMap
are defined
by looking into keySet() and values().

- Fixed a few methods of EMPTY_LIST which were throwing exception
semantically (see the introduction).

- The interval iterators are now list iterators, except for longs.

- Fixed a bug in size() for array lists (reducing the size of an array
would lead to an exception).

- Fixed double bug in hash tables: first of all, on very small sizes adding
growthFactor would have left the size unchanged, giving rise to infinite
loops. (Thanks to Heikki Uusitalo for reporting this bug.) Second, growthFactor
was not being used *at all* by hash maps.

- Fixed entries emitted by singleton maps. Now they are type-specific.

- Fixed a number of minor glitches in gencsource.sh, and added some comments.

- HashCommon.removed has been renamed HashCommon.REMOVED.

- Boolean objects are now generated using valueOf() instead of the constructor.

- New type-specific wrappers for list iterators.

3.0 -> 3.1

- IMPORTANT: it.unimi.dsi.fastutil.Iterators methods
have
been spread in type-specific static containers.

- New Stack interface, implemented by type-specific lists.

- New static container classes Collections, Sets, and Lists.
Presently they just provide empty containers.

- New type-specific static contains (e.g., IntSets) providing
singletons and synchronized wrappers.

- Entry sets now have entries that are equal() to entries
coming from corresponding maps in java.util.

- Spelling everywhere changed to Pure American. "synchronized" in code and "synchronise" in text side-by-side were looking really wierd...

3.0 -> 3.0.1

- New unwrap() methods for type-specific collections.

- Fixed old-as-world-bug, apparently wide but that evidently no one ever noticed: AbstractMap was not serialisable, and, as a result, the default return value was not serialised (I find sincerely counterintuitive that making a class serialisable doesn't do the same for its supertypes). It wasn't ever even *documented* as preserved, so probably everyone thought
this was my idea, too. Too bad this breaks once more serialisation compatibility. Since I had to break some serialisation anyway, I decided to eliminate the residual serialisation of p in hash table classes, too (which breaks serialisation for all hash-based classes).

2.60 -> 3.0

- IMPORTANT: All classes have been repackaged following the type of elements/keys. Sources will have to be retouched (just to change the import clause) and recompiled.

- IMPORTANT: Because of an unavoidable name clash in the new type-specific list interface, the method remove(int) of IntCollection has been renamed rem(int). The only really unpleasant effect is that you must use rem(int) on variables of type IntCollection that are not of type IntSet (as IntSet reinstates remove(int) in its right place)--for instance, IntList.

- Brand-new implementation of type-specific lists, with all the features you'd expect and more.

- Insertions for readObject() in hash tables are now handled in a special way (20% faster).

- Implemented linear-time tree reconstruction for readObject() (in practice, more than twice faster).

- Fixed a problem with serialisation of hash tables: the table would have been reloaded with the same p, even if it was preposterous. We still save p, however, to avoid breaking serialisation compatibility.

- Fixed missing implementation of type-specific sets, which should have extended type-specific collections, but they weren't.

- The default return value is now protected.

- New family of pour() methods that pour an iterator into a set.

- New programmable growth factor for hash-table-based classes.

- Eliminated a few useless method calls in tree map.

- Wide range of complex assertions, which are compiled in or out using the
"private static final boolean" idiom.

- For references we now use System.identityHashCode(); this shouldn't change
much, but it seems definitely more sensible.

- Fixed major bug in subSet()/subMap():
creating a subMap of a tailMap (or
headMap) a right extreme (left, resp.) equal to 0 would have caused the
creation of a tailMap (or headMap, resp.), discarding the extreme. Very,
very unlikely, but it happened in a test.

- Fixed small bug in standard remove() method of submaps, which would have
returned a default return value wrapped in a suitable object instead of
null on non-existing keys.

2.52 -> 2.60

- IMPORTANT: Major overhaul of iterators. Now iterators must be skippable,
so previous implementation of type-specific iterator interfaces will
not work. However, new abstract classes allow to build iterator with ease
by providing for free the skipping logic, and many useful static methods in
Iterators allow to generate type-specific iterators wrapping standard
iterators, arrays, etc.

- Better strategy for clear() on hash tables: we don't do anything only
if all entries are free (which means that an empty table with deleted
entry will be cleared).

2.51
-> 2.52

- IMPORTANT: The package name has changed to it.unimi.dsi.fastutil to be
uniform with JPackage conventions. However, this means that you must manually
erase the old one and update your sources.

- clear() doesn't do anything on empty hash tables.

2.50 -> 2.51
- New trim(int) method to reduce a hash table size avoiding to

make it too small.

- serialVersionUID is now fixed, to avoid future incompatibilities.

2.11 -> 2.50

- IMPORTANT: The Collection interface now prescribes an iterator
method with a type-specific name (e.g., intIterator()) that
returns directly a type-specific iterator.

- New Reference maps and sets that allow to store more quickly
canonised objects.

- New linked maps mimicking java.util's, but with a boatload
of additional features.

- Small bug fix: the get(Object) method would return null
instead of the default return value for maps with object
keys.

- Major bug fix: iterating backwards on submaps was leading
to unpredictable
results.

- Major bug fix: cloning maps would have caused inconsistent behaviour.

- Major code redistribution: now whenever possible wrappers
belong to abstract superclasses.

2.1 -> 2.11

- Now we cache the hash of an object before entering
the hash table loop.

2.0 -> 2.1

- A simple optimisation in hash-table inner loops has given
quite a performance boost under certain conditions (we do
not compute the secondary hashing if it is not necessary). Inspired
by Gnu Trove.
- The trim() method would have in fact trimmed nothing, just
rehashed the table.
- The computed maxFill value was sligtly too small.
- Also tree sets now have constructors from arrays.
- More internal methods have been made final.

1.3 -> 2.0

- ALL MAPS AND SETS HAVE NEW NAMES DEPENDING ON THE IMPLEMENTATION.
- Introducing new high-performance, low memory-footprint implementation of
 SortedMap and SortedSet.
- Two tree implementations are available: RB trees and
 AVL trees. Both implementations are threaded.
 See the README.
- Fixed a bug in hashCode() and contains() for HashMap.drv (it was
 considering keys only!).
- Fixed a bug in contains() for entrySet() in all maps (it was using
 VALUE_EQUAL to test equality for values given as objects).
- I realised that a default return value can be useful also for maps and sets
 returning objects, so now you have it. It is even independent for submaps and
 subsets.
- Classes are no longer final. The performance gain is around 1%, and
 the decrease in usefulness is orders of magnitudes greater.
- We now check equality using first hashCode() and then equals().
- The tests for speed now warm up the trees by doing repeated insertions
 and deletions, so that the benefits of a better balancing criterion
 are more evident.
- The regression tests are much more stringent.
- Fixed hashCode() for hash maps (wasn't conforming to the Map interface
 specification).
- Implemented linear cloning for tree classes.

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/CHANGES
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package PACKAGE;

import it.unimi.dsi.fastutil.objects.Object2IntOpenHashMap;
```

```
import java.io.Serializable;
import java.util.concurrent.locks.ReentrantReadWriteLock;
import java.util.concurrent.locks.ReentrantReadWriteLock.ReadLock;
import java.util.concurrent.locks.ReentrantReadWriteLock.WriteLock;

/** A concurrent counting
 map. The map is made by a number of <em>stripes</em> (instances of {@link Object2IntOpenHashMap})
* which are accessed independently
* using a {@link ReentrantReadWriteLock}. Only one thread can write in a stripe at a time, but different stripes
* can be modified independently and read access can happen concurrently on each stripe.
*
* @param <K> the type of keys.
*/

public class STRIPED_OPEN_HASH_MAP KEY_VALUE_GENERIC extends ABSTRACT_MAP
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable {
 private static final long serialVersionUID = 1L;

 /** The stripes. Keys are distributed among them using the lower bits of their {@link Object#hashCode()}. */
 private final OPEN_HASH_MAP KEY_VALUE_GENERIC[] map;
 /** An array of locks parallel to {@link #map}, protecting each stripe. */
 private final transient ReentrantReadWriteLock[] lock;
 /** {@link #map map.length} &minus; 1, cached. */
 private final int mask;

 /** Creates a new concurrent counting map with concurrency
 level equal to {@link Runtime#availableProcessors()}. */
 public STRIPED_OPEN_HASH_MAP() {
  this(Runtime.getRuntime().availableProcessors());
 }

 /** Creates a new concurrent counting map.
  *
  * @param concurrencyLevel the number of stripes (it will be {@linkplain Integer#highestOneBit(int) forced to be a
power of two}); ideally, as large as the number of threads that will ever access
  * this map, but higher values require more space.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public STRIPED_OPEN_HASH_MAP(final int concurrencyLevel) {
  map = new OPEN_HASH_MAP[Integer.highestOneBit(concurrencyLevel)];
  lock = new ReentrantReadWriteLock[map.length];
  for(int i = map.length; i-- != 0;) {
   map[i] = new OPEN_HASH_MAP KEY_VALUE_GENERIC_DIAMOND();
   lock[i] = new ReentrantReadWriteLock();
  }
  mask = map.length - 1;
 }
```

```
#if KEYS_PRIMITIVE

 public VALUE_GENERIC_CLASS get(final KEY_CLASS k) {
  final int stripe = KEY2INTHASH(k) & mask;
  final ReadLock readLock = lock[stripe].readLock();
  try
  {
   readLock.lock();
   return map[stripe].get(k);
  }
  finally {
   readLock.unlock();
  }
 }

#endif

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
  final int stripe = KEY2INTHASH(k) & mask;
  final ReadLock readLock = lock[stripe].readLock();
  try {
   readLock.lock();
   return map[stripe].GET_VALUE(k);
  }
  finally {
   readLock.unlock();
  }
 }

 public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
  final int stripe = KEY2INTHASH(k) & mask;
  final WriteLock writeLock = lock[stripe].writeLock();
  try {
   writeLock.lock();
   return map[stripe].put(k, v);
  }
  finally {
   writeLock.unlock();
  }
 }

 public VALUE_GENERIC_TYPE putIfAbsent(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE
v) {
  final int stripe = KEY2INTHASH(k) & mask;
  final WriteLock writeLock = lock[stripe].writeLock();
  try {
```

```
  writeLock.lock();
  if (map[stripe].containsKey(k)) return map[stripe].get(k);
  return
 map[stripe].put(k, v);
  }
  finally {
  writeLock.unlock();
  }
 }

#if VALUES_PRIMITIVE || KEYS_PRIMITIVE

 public VALUE_GENERIC_CLASS put(final KEY_GENERIC_CLASS ok, final VALUE_GENERIC_CLASS ov)
{
  final int stripe = KEY2INTHASH(ok) & mask;
  final WriteLock writeLock = lock[stripe].writeLock();
  try {
  writeLock.lock();
  return map[stripe].put(ok, ov);
  }
  finally {
  writeLock.unlock();
  }
 }


 public VALUE_GENERIC_CLASS putIfAbsent(final KEY_GENERIC_CLASS ok, final
VALUE_GENERIC_CLASS ov) {
  final int stripe = KEY2INTHASH(ok) & mask;
  final WriteLock writeLock = lock[stripe].writeLock();
  try {
  writeLock.lock();
  if (map[stripe].containsKey(ok)) return map[stripe].get(ok);
  return map[stripe].put(ok, ov);
  }
  finally {
  writeLock.unlock();
  }
 }

#endif

 public int size() {
  int size = 0;
  for(int stripe = lock.length; stripe-- != 0;) {
  final ReadLock readLock = lock[stripe].readLock();
  try {
   readLock.lock();
```

```
    size += map[stripe].size();
   }
   finally
  {
    readLock.unlock();
   }
  }

  return size;
 }

 public FastEntrySet KEY_VALUE_GENERIC ENTRYSET() {
  throw new UnsupportedOperationException();
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/StripedOpenHashMap.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import it.unimi.dsi.fastutil.objects.ObjectIterator;
import it.unimi.dsi.fastutil.objects.ObjectIterable;
import it.unimi.dsi.fastutil.objects.ObjectSet;
import it.unimi.dsi.fastutil.objects.ObjectSets;

import VALUE_PACKAGE.VALUE_COLLECTION;
import VALUE_PACKAGE.VALUE_COLLECTIONS;
#if ! VALUE_CLASS_Object
```

```
import
 VALUE_PACKAGE.VALUE_SETS;
#endif

import java.util.Map;
import java.util.function.Consumer;
import PACKAGE.MAP.FastEntrySet;

/** A class providing static methods and objects that do useful things with type-specific maps.
*
* @see java.util.Collections
*/

public final class MAPS {

 private MAPS() {}

 /** Returns an iterator that will be {@linkplain FastEntrySet fast}, if possible, on the {@linkplain Map#entrySet()
entry set} of the provided {@code map}.
  * @param map a map from which we will try to extract a (fast) iterator on the entry set.
  * @return an iterator on the entry set of the given map that will be fast, if possible.
  * @since 8.0.0
  */
 SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
 public static KEY_VALUE_GENERIC ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> fastIterator(MAP
KEY_VALUE_GENERIC map) {
  final ObjectSet<MAP.Entry KEY_VALUE_GENERIC> entries = map.ENTRYSET();
  return entries instanceof MAP.FastEntrySet ? ((MAP.FastEntrySet KEY_VALUE_GENERIC) entries).fastIterator()
 : entries.iterator();
 }

 /** Iterates {@linkplain FastEntrySet#fastForEach(Consumer) quickly}, if possible, on the {@linkplain
Map#entrySet() entry set} of the provided {@code map}.
  * @param map a map on which we will try to iterate {@linkplain FastEntrySet#fastForEach(Consumer) quickly}.
  * @param consumer the consumer that will be passed to  {@link FastEntrySet#fastForEach(Consumer)}, if
possible, or to {@link Iterable#forEach(Consumer)}.
  * @since 8.1.0
  */
 SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
 public static KEY_VALUE_GENERIC void fastForEach(MAP KEY_VALUE_GENERIC map, final Consumer<?
super MAP.Entry KEY_VALUE_GENERIC> consumer) {
  final ObjectSet<MAP.Entry KEY_VALUE_GENERIC> entries = map.ENTRYSET();
  if (entries instanceof MAP.FastEntrySet) ((MAP.FastEntrySet KEY_VALUE_GENERIC)
entries).fastForEach(consumer);
  else entries.forEach(consumer);
 }

 /** Returns an iterable yielding an iterator that will be {@linkplain FastEntrySet fast}, if possible,
```

```
 on the {@linkplain Map#entrySet() entry set} of the provided {@code map}.
  * @param map a map from which we will try to extract an iterable yielding a (fast) iterator on the entry set.
  * @return an iterable  yielding an iterator on the entry set of the given map that will be
  * fast, if possible.
  * @since 8.0.0
  */
 SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
 public static KEY_VALUE_GENERIC ObjectIterable<MAP.Entry KEY_VALUE_GENERIC> fastIterable(MAP
KEY_VALUE_GENERIC map) {
  final ObjectSet<MAP.Entry KEY_VALUE_GENERIC> entries = map.ENTRYSET();
  return entries instanceof MAP.FastEntrySet ? new ObjectIterable<MAP.Entry KEY_VALUE_GENERIC>() {
   public ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() { return ((MAP.FastEntrySet
KEY_VALUE_GENERIC)entries).fastIterator(); }
   public void forEach(final Consumer<? super MAP.Entry KEY_VALUE_GENERIC> consumer) {
((MAP.FastEntrySet KEY_VALUE_GENERIC)entries).fastForEach(consumer); }
  } : entries;
 }

 /** An immutable class representing
 an empty type-specific map.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific map.
  */

 public static class EmptyMap KEY_VALUE_GENERIC extends FUNCTIONS.EmptyFunction
KEY_VALUE_GENERIC implements MAP KEY_VALUE_GENERIC, java.io.Serializable, Cloneable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected EmptyMap() {}

  @Override
  public boolean containsValue(final VALUE_TYPE v) { return false; }

#if VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public boolean containsValue(final Object ov) { return false; }
#endif

  @Override
  public void putAll(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> m) {
throw new UnsupportedOperationException(); }

  @SuppressWarnings("unchecked")
```

```
 @Override
 public ObjectSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { return ObjectSets.EMPTY_SET;
}

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 @Override
 public SET KEY_GENERIC keySet() { return SETS.EMPTY_SET; }

 SUPPRESS_WARNINGS_VALUE_UNCHECKED
 @Override
 public VALUE_COLLECTION VALUE_GENERIC values() { return VALUE_SETS.EMPTY_SET; }

 @Override
 public Object clone() { return EMPTY_MAP; }

 @Override
 public boolean isEmpty() { return true; }

 @Override
 public int hashCode() { return 0; }

 @Override
 public boolean equals(final Object o) {
  if (! (o instanceof Map)) return false;
  return ((Map<?,?>)o).isEmpty();
 }

 @Override
 public String toString() { return "{}"; }
}


/** An empty type-specific map (immutable). It is serializable and cloneable.
 */
SUPPRESS_WARNINGS_KEY_VALUE_RAWTYPES
public static final EmptyMap EMPTY_MAP = new EmptyMap();

#if KEYS_REFERENCE || VALUES_REFERENCE
/** Returns an empty map (immutable). It is serializable and cloneable.
 *
 * <p>This method provides a typesafe access to {@link #EMPTY_MAP}.
 * @return an empty map (immutable).

 */
@SuppressWarnings("unchecked")
public static KEY_VALUE_GENERIC MAP KEY_VALUE_GENERIC emptyMap() {
 return EMPTY_MAP;
}
```

```
#endif

 /** An immutable class representing a type-specific singleton map.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific map.
  */

 public static class Singleton KEY_VALUE_GENERIC extends FUNCTIONS.Singleton KEY_VALUE_GENERIC
implements MAP KEY_VALUE_GENERIC, java.io.Serializable, Cloneable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected transient ObjectSet<MAP.Entry KEY_VALUE_GENERIC> entries;
  protected transient SET KEY_GENERIC keys;
  protected transient VALUE_COLLECTION VALUE_GENERIC values;

  protected Singleton(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value) {
   super(key, value);
  }

  @Override
  public boolean containsValue(final VALUE_TYPE v) { return VALUE_EQUALS(value, v); }
#if VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated
 Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public boolean containsValue(final Object ov) { return VALUE_EQUALS(VALUE_OBJ2TYPE(ov), value); }
#endif

  @Override
  public void putAll(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> m) {
throw new UnsupportedOperationException(); }

  @Override
  public ObjectSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { if (entries == null) entries =
ObjectSets.singleton(new ABSTRACT_MAP.BasicEntry KEY_VALUE_GENERIC_DIAMOND(key, value));
return entries; }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
#else
  /** {@inheritDoc} */
#endif
```

```
  @Override
  @SuppressWarnings({ "rawtypes", "unchecked" })
  public ObjectSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
(ObjectSet)ENTRYSET(); }

  @Override
  public SET KEY_GENERIC keySet() { if (keys == null) keys
 = SETS.singleton(key); return keys; }

  @Override
  public VALUE_COLLECTION VALUE_GENERIC values() { if (values == null) values =
VALUE_SETS.singleton(value); return values; }

  @Override
  public boolean isEmpty() { return false; }

  @Override
  public int hashCode() { return KEY2JAVAHASH(key) ^ VALUE2JAVAHASH(value); }

  @Override
  public boolean equals(final Object o) {
   if (o == this) return true;
   if (! (o instanceof Map)) return false;

   Map<?,?> m = (Map<?,?>)o;
   if (m.size() != 1) return false;
   return m.entrySet().iterator().next().equals(entrySet().iterator().next());
  }

  @Override
  public String toString() { return "{" + key + "=>" + value + "}"; }
 }

 /** Returns a type-specific immutable map containing only the specified pair. The returned map is serializable and
cloneable.
  *
  * <p>Note that albeit the returned map is immutable, its default return value may be changed.
  *
  * @param key the only key of the returned map.
  * @param value the only value
 of the returned map.
  * @return a type-specific immutable map containing just the pair {@code &lt;key,value&gt;}.
  */

 public static KEY_VALUE_GENERIC MAP KEY_VALUE_GENERIC singleton(final KEY_GENERIC_TYPE
key, VALUE_GENERIC_TYPE value) { return new Singleton KEY_VALUE_GENERIC_DIAMOND(key, value);
 }
```

```
#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

 /** Returns a type-specific immutable map containing only the specified pair. The returned map is serializable and cloneable.
  *
  * <p>Note that albeit the returned map is immutable, its default return value may be changed.
  *
  * @param key the only key of the returned map.
  * @param value the only value of the returned map.
  * @return a type-specific immutable map containing just the pair {@code &lt;key,value&gt;}.
  */

 public static KEY_VALUE_GENERIC MAP KEY_VALUE_GENERIC singleton(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS value) { return new Singleton KEY_VALUE_GENERIC_DIAMOND(KEY_CLASS2TYPE(key), VALUE_CLASS2TYPE(value));
 }

#endif


 /** A synchronized wrapper class for maps. */

 public static class SynchronizedMap KEY_VALUE_GENERIC extends FUNCTIONS.SynchronizedFunction KEY_VALUE_GENERIC implements MAP KEY_VALUE_GENERIC, java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final MAP KEY_VALUE_GENERIC map;

  protected transient ObjectSet<MAP.Entry KEY_VALUE_GENERIC> entries;
  protected transient SET KEY_GENERIC keys;
  protected transient VALUE_COLLECTION VALUE_GENERIC values;

  protected SynchronizedMap(final MAP KEY_VALUE_GENERIC m, final Object sync) {
   super(m, sync);
   this.map = m;
  }

  protected SynchronizedMap(final MAP KEY_VALUE_GENERIC m) {
   super(m);
   this.map = m;
  }

  @Override
  public boolean containsValue(final VALUE_TYPE v) { synchronized(sync) { return map.containsValue(v); } }

#if VALUES_PRIMITIVE
  /** {@inheritDoc}
```

```
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public
 boolean containsValue(final Object ov) { synchronized(sync) { return map.containsValue(ov); } }
#endif

  @Override
  public void putAll(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> m) {
synchronized(sync) { map.putAll(m); } }

  @Override
  public ObjectSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { synchronized(sync) { if (entries == null)
entries = ObjectSets.synchronize(map.ENTRYSET(), sync); return entries; } }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
#else
  /** {@inheritDoc} */
#endif
  @Override
  @SuppressWarnings({ "unchecked", "rawtypes" })
  public ObjectSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
(ObjectSet)ENTRYSET(); }

  @Override
  public SET KEY_GENERIC keySet() {
   synchronized(sync) { if (keys == null) keys = SETS.synchronize(map.keySet(), sync); return keys; }
  }

  @Override
  public
 VALUE_COLLECTION VALUE_GENERIC values() {
   synchronized(sync) { if (values == null) return VALUE_COLLECTIONS.synchronize(map.values(), sync); return
values; }
  }

  @Override
  public boolean isEmpty() { synchronized(sync) { return map.isEmpty(); } }

  @Override
  public int hashCode() { synchronized(sync) { return map.hashCode(); } }

  @Override
  public boolean equals(final Object o) {
   if (o == this) return true;
```

```
  synchronized(sync) { return map.equals(o); }
  }

  private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
   synchronized(sync) { s.defaultWriteObject(); }
  }

  // Defaultable methods

  @Override
  public VALUE_GENERIC_TYPE getOrDefault(final KEY_TYPE key, final VALUE_GENERIC_TYPE
defaultValue) { synchronized(sync) { return map.getOrDefault(key, defaultValue); } }

  @Override
  public void forEach(final java.util.function.BiConsumer<? super KEY_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS> action) { synchronized (sync) { map.forEach(action);
 } }

  @Override
  public void replaceAll(final java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> function) { synchronized (sync) {
map.replaceAll(function); } }

  @Override
  public VALUE_GENERIC_TYPE putIfAbsent(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value) { synchronized(sync) { return map.putIfAbsent(key, value); } }

  @Override
  public boolean remove(final KEY_TYPE key, final VALUE_TYPE value) { synchronized(sync) { return
map.remove(key, value); } }

  @Override
  public VALUE_GENERIC_TYPE replace(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value) { synchronized(sync) { return map.replace(key, value); } }

  @Override
  public boolean replace(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE oldValue, final
VALUE_GENERIC_TYPE newValue) { synchronized(sync) { return map.replace(key, oldValue, newValue); } }

#ifdef JDK_PRIMITIVE_FUNCTION
  @Override
  public VALUE_GENERIC_TYPE
 COMPUTE_IF_ABSENT_JDK(final KEY_GENERIC_TYPE key, final JDK_PRIMITIVE_FUNCTION
KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) { synchronized(sync) { return
map.COMPUTE_IF_ABSENT_JDK(key, mappingFunction); } }
#endif

#if KEYS_PRIMITIVE && VALUES_PRIMITIVE
```

```
  @Override
  public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_NULLABLE(final KEY_GENERIC_TYPE key,
final JDK_KEY_TO_GENERIC_FUNCTION<? extends VALUE_GENERIC_CLASS> mappingFunction) {
synchronized(sync) { return map.COMPUTE_IF_ABSENT_NULLABLE(key, mappingFunction); } }
#endif

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
  @Override
  public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_PARTIAL(final KEY_GENERIC_TYPE key, final
FUNCTION KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) { synchronized(sync) {
return map.COMPUTE_IF_ABSENT_PARTIAL(key, mappingFunction); } }
#endif

  @Override
  public VALUE_GENERIC_TYPE COMPUTE_IF_PRESENT(final KEY_GENERIC_TYPE key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super
 VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) {
   synchronized (sync) { return map.COMPUTE_IF_PRESENT(key, remappingFunction); }
  }

  @Override
  public VALUE_GENERIC_TYPE COMPUTE(final KEY_GENERIC_TYPE key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) {
   synchronized (sync) { return map.COMPUTE(key, remappingFunction); }
  }

  @Override
  public VALUE_GENERIC_TYPE MERGE(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value, final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) {
   synchronized (sync) { return map.MERGE(key, value, remappingFunction); }
  }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS getOrDefault(final
 Object key, final VALUE_GENERIC_CLASS defaultValue) { synchronized (sync) { return map.getOrDefault(key,
defaultValue); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public boolean remove(final Object key, final Object value) { synchronized (sync) { return map.remove(key, value);
```

```
} }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS replace(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value) { synchronized (sync) { return map.replace(key, value); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public boolean replace(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS oldValue, final
VALUE_GENERIC_CLASS newValue) { synchronized (sync) { return map.replace(key, oldValue, newValue); } }

  /** {@inheritDoc}

 * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS putIfAbsent(final KEY_GENERIC_CLASS key, final
VALUE_GENERIC_CLASS value) { synchronized (sync) { return map.putIfAbsent(key, value); } }

#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
#endif
  @Override
  public VALUE_GENERIC_CLASS computeIfAbsent(final KEY_GENERIC_CLASS key, final
java.util.function.Function<? super KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS>
mappingFunction) { synchronized (sync) { return map.computeIfAbsent(key, mappingFunction); } }

#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
#endif
  @Override
  public VALUE_GENERIC_CLASS computeIfPresent(final KEY_GENERIC_CLASS key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS,
 ? extends VALUE_GENERIC_CLASS> remappingFunction) { synchronized (sync) { return
map.computeIfPresent(key, remappingFunction); } }

#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
```

```
  @Deprecated
#endif
  @Override
  public VALUE_GENERIC_CLASS compute(final KEY_GENERIC_CLASS key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) { synchronized (sync) { return map.compute(key,
remappingFunction); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS merge(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value, final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) { synchronized (sync)
{ return map.merge(key, value, remappingFunction);
 } }
#endif

 }

 /** Returns a synchronized type-specific map backed by the given type-specific map.
  *
  * @param m the map to be wrapped in a synchronized map.
  * @return a synchronized view of the specified map.
  * @see java.util.Collections#synchronizedMap(Map)
  */
 public static KEY_VALUE_GENERIC MAP KEY_VALUE_GENERIC synchronize(final MAP
KEY_VALUE_GENERIC m) { return new SynchronizedMap KEY_VALUE_GENERIC_DIAMOND(m); }

 /** Returns a synchronized type-specific map backed by the given type-specific map, using an assigned object to
synchronize.
  *
  * @param m the map to be wrapped in a synchronized map.
  * @param sync an object that will be used to synchronize the access to the map.
  * @return a synchronized view of the specified map.
  * @see java.util.Collections#synchronizedMap(Map)
  */

 public static KEY_VALUE_GENERIC MAP KEY_VALUE_GENERIC synchronize(final MAP
KEY_VALUE_GENERIC m, final Object sync) { return new SynchronizedMap
KEY_VALUE_GENERIC_DIAMOND(m,
 sync); }



 /** An unmodifiable wrapper class for maps. */
```

```
 public static class UnmodifiableMap KEY_VALUE_GENERIC extends FUNCTIONS.UnmodifiableFunction
KEY_VALUE_GENERIC implements MAP KEY_VALUE_GENERIC, java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final MAP KEY_VALUE_GENERIC map;

  protected transient ObjectSet<MAP.Entry KEY_VALUE_GENERIC> entries;
  protected transient SET KEY_GENERIC keys;
  protected transient VALUE_COLLECTION VALUE_GENERIC values;

  protected UnmodifiableMap(final MAP KEY_VALUE_GENERIC m) {
   super(m);
   this.map = m;
  }

  @Override
  public boolean containsValue(final VALUE_TYPE v) { return map.containsValue(v); }
#if VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public boolean containsValue(final Object ov) { return map.containsValue(ov); }
#endif

  @Override
  public void putAll(final Map<?
 extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> m) { throw new
UnsupportedOperationException(); }

  @Override
  public ObjectSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { if (entries == null) entries =
ObjectSets.unmodifiable(map.ENTRYSET()); return entries; }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
#else
  /** {@inheritDoc} */
#endif
  @Override
  @SuppressWarnings({ "unchecked", "rawtypes" })
  public ObjectSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
(ObjectSet)ENTRYSET(); }

  @Override
```

```
  public SET KEY_GENERIC keySet() { if (keys == null) keys = SETS.unmodifiable(map.keySet()); return keys; }

  @Override
  public VALUE_COLLECTION VALUE_GENERIC values() { if (values == null) return
VALUE_COLLECTIONS.unmodifiable(map.values()); return values; }

  @Override
  public boolean isEmpty() { return map.isEmpty(); }

  @Override
  public
 int hashCode() { return map.hashCode(); }

  @Override
  public boolean equals(final Object o) {
   if (o == this) return true;
   return map.equals(o);
  }

  // Defaultable methods

  @Override
  public VALUE_GENERIC_TYPE getOrDefault(final KEY_TYPE key, final VALUE_GENERIC_TYPE
defaultValue) { return map.getOrDefault(key, defaultValue); }

  @Override
  public void forEach(final java.util.function.BiConsumer<? super KEY_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS> action) { map.forEach(action); }

  @Override
  public void replaceAll(final java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> function) { throw new
UnsupportedOperationException(); }

  @Override
  public VALUE_GENERIC_TYPE putIfAbsent(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value) { throw new UnsupportedOperationException(); }

  @Override
  public boolean remove(final KEY_TYPE key, final VALUE_TYPE value) { throw new
UnsupportedOperationException();
 }

  @Override
  public VALUE_GENERIC_TYPE replace(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value) { throw new UnsupportedOperationException(); }

  @Override
```

```
  public boolean replace(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE oldValue, final
VALUE_GENERIC_TYPE newValue) { throw new UnsupportedOperationException(); }

#ifdef JDK_PRIMITIVE_FUNCTION
  @Override
  public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_JDK(final KEY_GENERIC_TYPE key, final
JDK_PRIMITIVE_FUNCTION KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) {
throw new UnsupportedOperationException(); }
#endif

#if KEYS_PRIMITIVE && VALUES_PRIMITIVE
  @Override
  public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_NULLABLE(final KEY_GENERIC_TYPE key,
final JDK_KEY_TO_GENERIC_FUNCTION<? extends VALUE_GENERIC_CLASS> mappingFunction) { throw
new UnsupportedOperationException(); }
#endif

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
  @Override
  public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_PARTIAL(final KEY_GENERIC_TYPE
 key, final FUNCTION KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) { throw new
UnsupportedOperationException(); }
#endif

  @Override
  public VALUE_GENERIC_TYPE COMPUTE_IF_PRESENT(final KEY_GENERIC_TYPE key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) { throw new UnsupportedOperationException(); }

  @Override
  public VALUE_GENERIC_TYPE COMPUTE(final KEY_GENERIC_TYPE key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) { throw new UnsupportedOperationException(); }

  @Override
  public VALUE_GENERIC_TYPE MERGE(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value, final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) { throw new
UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
 || VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS getOrDefault(final Object key, final VALUE_GENERIC_CLASS
defaultValue) { return map.getOrDefault(key, defaultValue); }
```

```
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public boolean remove(final Object key, final Object value) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS replace(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public boolean replace(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
 oldValue, final VALUE_GENERIC_CLASS newValue) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS putIfAbsent(final KEY_GENERIC_CLASS key, final
VALUE_GENERIC_CLASS value) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
#endif
  @Override
  public VALUE_GENERIC_CLASS computeIfAbsent(final KEY_GENERIC_CLASS key, final
java.util.function.Function<? super KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS>
mappingFunction) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
#endif
  @Override
  public VALUE_GENERIC_CLASS computeIfPresent(final KEY_GENERIC_CLASS key, final
java.util.function.BiFunction<?
 super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS>
remappingFunction) { throw new UnsupportedOperationException(); }
```

```
#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
#endif
  @Override
  public VALUE_GENERIC_CLASS compute(final KEY_GENERIC_CLASS key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS merge(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value, final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) { throw new
UnsupportedOperationException(); }
#endif

 }

 /**
 Returns an unmodifiable type-specific map backed by the given type-specific map.
  *
  * @param m the map to be wrapped in an unmodifiable map.
  * @return an unmodifiable view of the specified map.
  * @see java.util.Collections#unmodifiableMap(Map)
  */
 public static KEY_VALUE_GENERIC MAP KEY_VALUE_GENERIC unmodifiable(final MAP
KEY_VALUE_GENERIC m) { return new UnmodifiableMap KEY_VALUE_GENERIC_DIAMOND(m); }

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Maps.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
```

```
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import static it.unimi.dsi.fastutil.BigArrays.copy;
import static it.unimi.dsi.fastutil.BigArrays.fill;
import static it.unimi.dsi.fastutil.BigArrays.set;

import it.unimi.dsi.fastutil.BigArrays;
import it.unimi.dsi.fastutil.Hash;
import it.unimi.dsi.fastutil.Size64;
import it.unimi.dsi.fastutil.HashCommon;
import
 static it.unimi.dsi.fastutil.HashCommon.bigArraySize;
import static it.unimi.dsi.fastutil.HashCommon.maxFill;

import java.util.Collection;
import java.util.Iterator;
import java.util.NoSuchElementException;


/**  A type-specific hash big set with with a fast, small-footprint implementation.
*
* <p>Instances of this class use a hash table to represent a big set: the number
* of elements in the set is limited only by the amount of core memory. The table
* (backed by a {@linkplain it.unimi.dsi.fastutil.BigArrays big array}) is
* filled up to a specified <em>load factor</em>, and then doubled in size to
* accommodate new entries. If the table is emptied below <em>one fourth</em>
* of the load factor, it is halved in size; however, the table is never reduced to a
* size smaller than that at creation time: this approach makes it
* possible to create sets with a large capacity in which insertions and
* deletions do not cause immediately rehashing. Moreover, halving
 is
* not performed when deleting entries from an iterator, as it would interfere
* with the iteration process.
*
* <p>Note that {@link #clear()} does not modify the hash table size.
* Rather, a family of {@linkplain #trim() trimming
* methods} lets you control the size of the table; this is particularly useful
* if you reuse instances of this class.
*
```

```
* <p>The methods of this class are about 30% slower than those of the corresponding non-big set.
*
* @see Hash
* @see HashCommon
*/

public class OPEN_HASH_BIG_SET KEY_GENERIC extends ABSTRACT_SET KEY_GENERIC implements
java.io.Serializable, Cloneable, Hash, Size64 {

 private static final long serialVersionUID = 0L;
 private static final boolean ASSERTS = ASSERTS_VALUE;

 /** The big array of keys. */
 protected transient KEY_GENERIC_TYPE[][] key;

 /** The mask for wrapping a position counter. */
 protected transient long mask;

 /** The mask for wrapping a segment counter. */
 protected transient int segmentMask;

 /** The
 mask for wrapping a base counter. */
 protected transient int baseMask;

 /** Whether this set contains the null key. */
 protected transient boolean containsNull;

 /** The current table size (always a power of 2). */
 protected transient long n;

 /** Threshold after which we rehash. It must be the table size times {@link #f}. */
 protected transient long maxFill;

 /** We never resize below this threshold, which is the construction-time {#n}. */
 protected final transient long minN;

 /** The acceptable load factor. */
 protected final float f;

 /** Number of entries in the set. */
 protected long size;


 /** Initialises the mask values. */
 private void initMasks() {
  mask = n - 1;
  /* Note that either we have more than one segment, and in this case all segments
```

```
  * are BigArrays.SEGMENT_SIZE long, or we have exactly one segment whose length
  * is a power of two. */
 segmentMask = key[0].length - 1;
 baseMask = key.length - 1;
}

/** Creates a new hash big set.
 *
 *
<p>The actual table size will be the least power of two greater than {@code expected}/{@code f}.
 *
 * @param expected the expected number of elements in the set.
 * @param f the load factor.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public OPEN_HASH_BIG_SET(final long expected, final float f) {
 if (f <= 0 || f > 1) throw new IllegalArgumentException("Load factor must be greater than 0 and smaller than or
equal to 1");
 if (n < 0) throw new IllegalArgumentException("The expected number of elements must be nonnegative");

 this.f = f;

 minN = n = bigArraySize(expected, f);
 maxFill = maxFill(n, f);
 key = KEY_GENERIC_BIG_ARRAY_CAST BIG_ARRAYS.newBigArray(n);
 initMasks();
}


/** Creates a new hash big set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor.
 *
 * @param expected the expected number of elements in the hash big set.
 */

public OPEN_HASH_BIG_SET(final long expected) {
 this(expected, DEFAULT_LOAD_FACTOR);
}

/** Creates a new hash big set with initial
expected {@link Hash#DEFAULT_INITIAL_SIZE} elements
 * and {@link Hash#DEFAULT_LOAD_FACTOR} as load factor.
 */

public OPEN_HASH_BIG_SET() {
 this(DEFAULT_INITIAL_SIZE, DEFAULT_LOAD_FACTOR);
}

/** Creates a new hash big set copying a given collection.
```

```
 *
 * @param c a {@link Collection} to be copied into the new hash big set.
 * @param f the load factor.
 */

public OPEN_HASH_BIG_SET(final Collection<? extends KEY_GENERIC_CLASS> c, final float f) {
 this(c.size(), f);
 addAll(c);
}

/** Creates a new hash big set  with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
 * copying a given collection.
 *
 * @param c a {@link Collection} to be copied into the new hash big set.
 */

public OPEN_HASH_BIG_SET(final Collection<? extends KEY_GENERIC_CLASS> c) {
 this(c, DEFAULT_LOAD_FACTOR);
}

/** Creates a new hash big set copying a given type-specific collection.
 *
 * @param c a type-specific collection to be copied into the new hash big set.
 * @param f the load factor.

*/

public OPEN_HASH_BIG_SET(final COLLECTION KEY_EXTENDS_GENERIC c, final float f) {
 this(c.size(), f);
 addAll(c);
}

/** Creates a new hash big set  with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
 * copying a given type-specific collection.
 *
 * @param c a type-specific collection to be copied into the new hash big set.
 */

public OPEN_HASH_BIG_SET(final COLLECTION KEY_EXTENDS_GENERIC c) {
 this(c, DEFAULT_LOAD_FACTOR);
}

/** Creates a new hash big set using elements provided by a type-specific iterator.
 *
 * @param i a type-specific iterator whose elements will fill the new hash big set.
 * @param f the load factor.
 */
```

```
 public OPEN_HASH_BIG_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i, final float f) {
  this(DEFAULT_INITIAL_SIZE, f);
  while(i.hasNext()) add(i.NEXT_KEY());
 }

 /** Creates a new hash big set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor using elements
provided by a type-specific iterator.
  *
  * @param i a type-specific
 iterator whose elements will fill the new hash big set.
  */

 public OPEN_HASH_BIG_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i) {
  this(i, DEFAULT_LOAD_FACTOR);
 }


#if KEYS_PRIMITIVE

 /** Creates a new hash big set using elements provided by an iterator.
  *
  * @param i an iterator whose elements will fill the new hash big set.
  * @param f the load factor.
  */

 public OPEN_HASH_BIG_SET(final Iterator<?> i, final float f) {
  this(ITERATORS.AS_KEY_ITERATOR(i), f);
 }
 /** Creates a new hash big set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor using elements
provided by an iterator.
  *
  * @param i an iterator whose elements will fill the new hash big set.
  */

 public OPEN_HASH_BIG_SET(final Iterator<?> i) {
  this(ITERATORS.AS_KEY_ITERATOR(i));
 }

#endif


 /** Creates a new hash big set and fills it with the elements of a given array.
  *
  * @param a an array whose elements will be used to fill the new hash big set.
  * @param offset the first element
 to use.
  * @param length the number of elements to use.
  * @param f the load factor.
```

```
 */

public OPEN_HASH_BIG_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length, final float f) {
 this(length < 0 ? 0 : length, f);
 ARRAYS.ensureOffsetLength(a, offset, length);
 for(int i = 0; i < length; i++) add(a[offset + i]);
}

/** Creates a new hash big set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor and fills it with the
elements of a given array.
 *
 * @param a an array whose elements will be used to fill the new hash big set.
 * @param offset the first element to use.
 * @param length the number of elements to use.
 */

public OPEN_HASH_BIG_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length) {
 this(a, offset, length, DEFAULT_LOAD_FACTOR);
}

/** Creates a new hash big set copying the elements of an array.
 *
 * @param a an array to be copied into the new hash big set.
 * @param f the load factor.
 */

public OPEN_HASH_BIG_SET(final
KEY_GENERIC_TYPE[] a, final float f) {
 this(a, 0, a.length, f);
}

/** Creates a new hash big set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
 * copying the elements of an array.
 *
 * @param a an array to be copied into the new hash big set.
 */

public OPEN_HASH_BIG_SET(final KEY_GENERIC_TYPE[] a) {
 this(a, DEFAULT_LOAD_FACTOR);
}

private long realSize() {
 return containsNull ? size - 1 : size;
}

private void ensureCapacity(final long capacity) {
 final long needed = bigArraySize(capacity, f);
 if (needed > n) rehash(needed);
```

```
 }

 @Override
 public boolean addAll(Collection<? extends KEY_GENERIC_CLASS> c) {
  final long size = c instanceof Size64 ? ((Size64)c).size64() : c.size();
  // The resulting collection will be at least c.size() big
  if (f <= .5) ensureCapacity(size); // The resulting collection will be sized for c.size() elements
  else ensureCapacity(size64() + size); // The resulting collection will be sized for size() + c.size() elements
  return
 super.addAll(c);
 }

#if KEYS_PRIMITIVE
 @Override
 public boolean addAll(COLLECTION c) {
  final long size = c instanceof Size64 ? ((Size64)c).size64() : c.size();
  if (f <= .5) ensureCapacity(size); // The resulting collection will be size for c.size() elements
  else ensureCapacity(size64() + size); // The resulting collection will be sized for size() + c.size() elements
  return super.addAll(c);
 }
#endif

 @Override
 public boolean add(final KEY_GENERIC_TYPE k) {
  int displ, base;

  if (KEY_IS_NULL(k)) {
   if (containsNull) return false;
   containsNull = true;
  }
  else {
   KEY_GENERIC_TYPE curr;
   final KEY_GENERIC_TYPE[][] key = this.key;
   final long h = KEY2LONGHASH(k);

   // The starting point.
   if (! KEY_IS_NULL(curr = key[base = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][displ = (int)(h &
segmentMask)])) {
    if (KEY_EQUALS_NOT_NULL(curr, k)) return false;
    while(! KEY_IS_NULL(curr = key[base = (base + ((displ = (displ + 1) & segmentMask) == 0 ?
 1 : 0)) & baseMask][displ]))
     if (KEY_EQUALS_NOT_NULL(curr, k)) return false;
   }

   key[base][displ] = k;
  }

  if (size++ >= maxFill) rehash(2 * n);
```

```
  if (ASSERTS) checkTable();
  return true;
 }

#if KEY_CLASS_Object
 /** Add a random element if not present, get the existing value if already present.
  *
  * This is equivalent to (but faster than) doing a:
  * <pre>
  * K exist = set.get(k);
  * if (exist == null) {
  *   set.add(k);
  *   exist = k;
  * }
  * </pre>
  */
 public KEY_GENERIC_TYPE addOrGet(final KEY_GENERIC_TYPE k) {
  int displ, base;

  if (KEY_IS_NULL(k)) {
   if (containsNull) return null;
   containsNull = true;
  }
  else {
   KEY_GENERIC_TYPE curr;
   final KEY_GENERIC_TYPE[][] key = this.key;
   final long h = KEY2LONGHASH(k);

   // The starting point.
   if (! KEY_IS_NULL(curr = key[base = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][displ = (int)(h &
segmentMask)])) {
    if (KEY_EQUALS_NOT_NULL(curr, k)) return curr;
    while(! KEY_IS_NULL(curr
 = key[base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0)) & baseMask][displ]))
     if (KEY_EQUALS_NOT_NULL(curr, k)) return curr;
   }

   key[base][displ] = k;
  }

  if (size++ >= maxFill) rehash(2 * n);
  if (ASSERTS) checkTable();
  return k;
 }
#endif

 /** Shifts left entries with the specified hash code, starting at the specified position,
  * and empties the resulting free entry.
```

```
 *
 * @param pos a starting position.
 */
protected final void shiftKeys(long pos) {
 // Shift entries with the same hash.
 long last, slot;
 final KEY_GENERIC_TYPE[][] key = this.key;

 for(;;) {
  pos = ((last = pos) + 1) & mask;

  for(;;) {
   if (KEY_IS_NULL(BigArrays.get(key, pos))) {
    set(key, last, KEY_NULL);
    return;
   }
   slot = KEY2LONGHASH(BigArrays.get(key, pos)) & mask;
   if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
   pos = (pos + 1) & mask;
  }

  set(key, last, BigArrays.get(key, pos));
 }
}

private boolean
removeEntry(final int base, final int displ) {
 size--;
 shiftKeys(base * (long)BigArrays.SEGMENT_SIZE + displ);
 if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
 return true;
}

private boolean removeNullEntry() {
 containsNull = false;
 size--;
 if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
 return true;
}

@Override
public boolean remove(final KEY_TYPE k) {
 if (KEY_IS_NULL(k)) {
  if (containsNull) return removeNullEntry();
  return false;
 }

 KEY_GENERIC_TYPE curr;
```

```
  final KEY_GENERIC_TYPE[][] key = this.key;
  final long h = KEY2LONGHASH(k);
  int displ, base;

  // The starting point.
  if (KEY_IS_NULL(curr = key[base = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][displ = (int)(h &
segmentMask)])) return false;
  if (KEY_EQUALS_NOT_NULL(curr, k)) return removeEntry(base, displ);
  while(true) {
   if (KEY_IS_NULL(curr = key[base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0))
 & baseMask][displ])) return false;
   if (KEY_EQUALS_NOT_NULL(curr, k)) return removeEntry(base, displ);
  }
 }

 @Override
 public boolean contains(final KEY_TYPE k) {
  if (KEY_IS_NULL(k)) return containsNull;

  KEY_GENERIC_TYPE curr;
  final KEY_GENERIC_TYPE[][] key = this.key;
  final long h = KEY2LONGHASH(k);
  int displ, base;

  // The starting point.
  if (KEY_IS_NULL(curr = key[base = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][displ = (int)(h &
segmentMask)])) return false;
  if (KEY_EQUALS_NOT_NULL(curr, k)) return true;
  while(true) {
   if (KEY_IS_NULL(curr = key[base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0)) &
baseMask][displ])) return false;
   if (KEY_EQUALS_NOT_NULL(curr, k)) return true;
  }
 }

#if KEY_CLASS_Object
 /** Returns the element of this set that is equal to the given key, or {@code null}.
  * @return the element of this set that is equal to the given key, or {@code null}.
  */
 public K get(final KEY_TYPE k) {
  if (k == null) return
 null; // This is correct independently of the value of containsNull

  KEY_GENERIC_TYPE curr;
  final KEY_GENERIC_TYPE[][] key = this.key;
  final long h = KEY2LONGHASH(k);
  int displ, base;
```

```
  // The starting point.
  if (KEY_IS_NULL(curr = key[base = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][displ = (int)(h &
segmentMask)])) return null;
  if (KEY_EQUALS_NOT_NULL(curr, k)) return curr;
  while(true) {
   if (KEY_IS_NULL(curr = key[base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0)) &
baseMask][displ])) return null;
   if (KEY_EQUALS_NOT_NULL(curr, k)) return curr;
  }
 }
#endif

 /* Removes all elements from this set.
  *
  */

 /** {@inheritDoc}
  *
  * <p>To increase object reuse, this method does not change the table size.
  * If you want to reduce the table size, you must use {@link #trim(long)}.
  */
 @Override
 public void clear() {
  if (size == 0) return;
  size = 0;
  containsNull = false;
  fill(key, KEY_NULL);
 }



 /** An iterator over a hash big
 set. */

 private class SetIterator implements KEY_ITERATOR KEY_GENERIC {
  /** The base of the last entry returned, if positive or zero; initially, the number of components
   of the key array. If negative, the last element returned was
   that of index {@code - base - 1} from the {@link #wrapped} list. */
  int base = key.length;
  /** The displacement of the last entry returned; initially, zero. */
  int displ;
  /** The index of the last entry that has been returned (or {@link Long#MIN_VALUE} if {@link #base} is
negative).
   It is -1 if either we did not return an entry yet, or the last returned entry has been removed. */
  long last = -1;
  /** A downward counter measuring how many entries must still be returned. */
  long c = size;
  /** A boolean telling us whether we should return the null key. */
```

```
 boolean mustReturnNull = OPEN_HASH_BIG_SET.this.containsNull;
 /** A lazily allocated list containing elements that have wrapped around the table because of removals. */
 ARRAY_LIST
KEY_GENERIC wrapped;

 @Override
 public boolean hasNext() { return c != 0; }

 @Override
 public KEY_GENERIC_TYPE NEXT_KEY() {
  if (! hasNext()) throw new NoSuchElementException();
  c--;

  if (mustReturnNull) {
   mustReturnNull = false;
   last = n;
   return KEY_NULL;
  }

  final KEY_GENERIC_TYPE[][] key = OPEN_HASH_BIG_SET.this.key;

  for(;;) {
   if (displ == 0 && base <= 0) {
    // We are just enumerating elements from the wrapped list.
    last = Long.MIN_VALUE;
    return wrapped.GET_KEY(- (--base) - 1);
   }

   if (displ-- == 0) displ = key[--base].length - 1;

   final KEY_GENERIC_TYPE k = key[base][displ];
   if (! KEY_IS_NULL(k)) {
    last = base * (long)BigArrays.SEGMENT_SIZE + displ;
    return k;
   }
  }
 }

 /** Shifts left entries with the specified hash code, starting at the specified position,
  * and empties the resulting free entry.
  *
  * @param pos a starting position.
  */
 private final void shiftKeys(long pos) {
  //
Shift entries with the same hash.
  long last, slot;
  KEY_GENERIC_TYPE curr;
```

```
  final KEY_GENERIC_TYPE[][] key = OPEN_HASH_BIG_SET.this.key;

  for(;;) {
   pos = ((last = pos) + 1) & mask;

   for(;;) {
    if(KEY_IS_NULL(curr = BigArrays.get(key, pos))) {
     set(key, last, KEY_NULL);
     return;
    }
    slot = KEY2LONGHASH(curr) & mask;
    if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
    pos = (pos + 1) & mask;
   }

   if (pos < last) { // Wrapped entry.
    if (wrapped == null) wrapped = new ARRAY_LIST KEY_GENERIC_DIAMOND();
    wrapped.add(BigArrays.get(key, pos));
   }

   set(key, last, curr);
  }
 }

 @Override
 public void remove() {
  if (last == -1) throw new IllegalStateException();
  if (last == n) OPEN_HASH_BIG_SET.this.containsNull = false;
  else if (base >= 0) shiftKeys(last);
  else {
   // We're removing wrapped entries.
#if KEYS_REFERENCE
   OPEN_HASH_BIG_SET.this.remove(wrapped.set(- base
 - 1, null));
#else
   OPEN_HASH_BIG_SET.this.remove(wrapped.GET_KEY(- base - 1));
#endif
   last = -1; // Note that we must not decrement size
   return;
  }

  size--;
  last = -1; // You can no longer remove this entry.
  if (ASSERTS) checkTable();
 }
}

@Override
```

```
public KEY_ITERATOR KEY_GENERIC iterator() {
 return new SetIterator();
}


/** Rehashes this set, making the table as small as possible.
 *
 * <p>This method rehashes the table to the smallest size satisfying the
 * load factor. It can be used when the set will not be changed anymore, so
 * to optimize access speed and size.
 *
 * <p>If the table size is already the minimum possible, this method
 * does nothing.
 *
 * @return true if there was enough memory to trim the set.
 * @see #trim(long)
 */

public boolean trim() {
 return trim(size);
}

/** Rehashes this set if the table is too large.
 *
 * <p>Let <var>N</var> be the smallest table size that can hold
 * <code>max(n,{@link #size64()})</code>
entries, still satisfying the load factor. If the current
 * table size is smaller than or equal to <var>N</var>, this method does
 * nothing. Otherwise, it rehashes this set in a table of size
 * <var>N</var>.
 *
 * <p>This method is useful when reusing sets.  {@linkplain #clear() Clearing a
 * set} leaves the table size untouched. If you are reusing a set
 * many times, you can call this method with a typical
 * size to avoid keeping around a very large table just
 * because of a few large transient sets.
 *
 * @param n the threshold for the trimming.
 * @return true if there was enough memory to trim the set.
 * @see #trim()
 */

public boolean trim(final long n) {
 final long l = bigArraySize(n, f);
 if (l >= this.n || size > maxFill(l, f)) return true;
 try {
  rehash(l);
 }
```

```
 catch(OutOfMemoryError cantDoIt) { return false; }
 return true;
}

/** Resizes the set.
 *
 * <p>This method implements the basic rehashing strategy, and may be
 * overriden
by subclasses implementing different rehashing strategies (e.g.,
 * disk-based rehashing). However, you should not override this method
 * unless you understand the internal workings of this class.
 *
 * @param newN the new size
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
protected void rehash(final long newN) {
 final KEY_GENERIC_TYPE key[][] = this.key;
 final KEY_GENERIC_TYPE newKey[][] = KEY_GENERIC_BIG_ARRAY_CAST
BIG_ARRAYS.newBigArray(newN);
 final long mask = newN - 1; // Note that this is used by the hashing macro
 final int newSegmentMask = newKey[0].length - 1;
 final int newBaseMask = newKey.length - 1;

 int base = 0, displ = 0, b, d;
 long h;
 KEY_GENERIC_TYPE k;

 for(long i = realSize(); i-- != 0;) {

  while(KEY_IS_NULL(key[base][displ])) base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0));

  k = key[base][displ];
  h = KEY2LONGHASH(k);

  // The starting point.
  if (! KEY_IS_NULL(newKey[b = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][d
= (int)(h & newSegmentMask)]))
   while(! KEY_IS_NULL(newKey[b = (b + ((d = (d + 1) & newSegmentMask) == 0 ? 1 : 0)) & newBaseMask][d]));

  newKey[b][d] = k;

  base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0));
 }

 this.n = newN;
 this.key = newKey;
 initMasks();
```

```
 maxFill = maxFill(n, f);
}

@Deprecated
@Override
public int size() {
 return (int)Math.min(Integer.MAX_VALUE, size);
}

@Override
public long size64() {
 return size;
}

@Override
public boolean isEmpty() {
 return size == 0;
}



/** Returns a deep copy of this big set.
 *
 * <p>This method performs a deep copy of this big hash set; the data stored in the
 * set, however, is not cloned. Note that this makes a difference only for object keys.
 *
 *  @return a deep copy of this big set.
 */
@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public OPEN_HASH_BIG_SET KEY_GENERIC clone() {
 OPEN_HASH_BIG_SET KEY_GENERIC c;
 try {
  c = (OPEN_HASH_BIG_SET KEY_GENERIC)super.clone();
 }
 catch(CloneNotSupportedException
cantHappen) {
  throw new InternalError();
 }
 c.key = copy(key);
 c.containsNull = containsNull;
 return c;
}

/** Returns a hash code for this set.
 *
 * This method overrides the generic method provided by the superclass.
 * Since {@code equals()} is not overriden, it is important
```

```
 * that the value returned by this method is the same value as
 * the one returned by the overriden method.
 *
 * @return a hash code for this set.
 */

@Override
public int hashCode() {
 final KEY_GENERIC_TYPE key[][] = this.key;
 int h = 0, base = 0, displ = 0;

 for(long j = realSize(); j-- != 0;) {
  while(KEY_IS_NULL(key[base][displ])) base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0));
#if KEYS_REFERENCE
  if (this != key[base][displ])
#endif
   h += KEY2JAVAHASH_NOT_NULL(key[base][displ]);
  base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0));
 }
 return h;
}


private void writeObject(java.io.ObjectOutputStream
s) throws java.io.IOException {
 final KEY_ITERATOR KEY_GENERIC i = iterator();
 s.defaultWriteObject();
 for(long j = size; j-- != 0;) s.WRITE_KEY(i.NEXT_KEY());
}


SUPPRESS_WARNINGS_KEY_UNCHECKED
private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
 s.defaultReadObject();

 n = bigArraySize(size, f);
 maxFill = maxFill(n, f);

 final KEY_GENERIC_TYPE[][] key = this.key = KEY_GENERIC_BIG_ARRAY_CAST
BIG_ARRAYS.newBigArray(n);

 initMasks();

 long h;
 KEY_GENERIC_TYPE k;
 int base, displ;

 for(long i = size; i-- != 0;) {
```

```
   k = KEY_GENERIC_CAST s.READ_KEY();

   if (KEY_IS_NULL(k)) containsNull = true;
   else {
    h = KEY2LONGHASH(k);
    if (! KEY_IS_NULL(key[base = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][displ = (int)(h &
segmentMask)]))
     while(! KEY_IS_NULL(key[base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0)) &
baseMask][displ]));
    key[base][displ] = k;
   }
  }

  if (ASSERTS)
 checkTable();
 }


#ifdef ASSERTS_CODE
 private void checkTable() {
  assert (n & -n) == n : "Table length is not a power of two: " + n;
  assert n == BigArrays.length(key);
  long n = this.n;
  while(n-- != 0)
   if (! KEY_IS_NULL(BigArrays.get(key, n)) && ! contains(BigArrays.get(key, n)))
    throw new AssertionError("Hash table has key " + BigArrays.get(key, n) + " marked as occupied, but the key does
not belong to the table");

#if KEYS_PRIMITIVE
  java.util.HashSet<KEY_GENERIC_CLASS> s = new java.util.HashSet<KEY_GENERIC_CLASS> ();
#else
  java.util.HashSet<Object> s = new java.util.HashSet<Object>();
#endif

  for(long i = size(); i-- != 0;)
   if (! KEY_IS_NULL(BigArrays.get(key, i)) && ! s.add(BigArrays.get(key, i))) throw new AssertionError("Key " +
BigArrays.get(key, i) + " appears twice");

 }
#else
 private void checkTable() {}
#endif

#ifdef TEST

 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);
```

```
 private
 static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif KEY_CLASS_Object
  return Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }


 private static final class ArrayComparator implements java.util.Comparator {
  public int compare(Object a, Object b) {
   byte[] aa = (byte[])a;
   byte[] bb = (byte[])b;
   int length = Math.min(aa.length, bb.length);
   for(int i = 0; i < length; i++) {
    if (aa[i] < bb[i]) return -1;
    if (aa[i] > bb[i]) return 1;
   }
   return aa.length == bb.length ? 0 : (aa.length < bb.length ? -1 : 1);
  }
 }

 private static final class MockSet extends java.util.TreeSet {
  private java.util.List list = new java.util.ArrayList();

  public MockSet(java.util.Comparator c) { super(c); }

  public boolean add(Object k) {
   if (! contains(k)) list.add(k);
   return super.add(k);
  }

  public
 boolean addAll(Collection c) {
   java.util.Iterator i = c.iterator();
   boolean result = false;
   while(i.hasNext()) result |= add(i.next());
   return result;
  }

  public boolean removeAll(Collection c) {
   java.util.Iterator i = c.iterator();
   boolean result = false;
```

```
 while(i.hasNext()) result |= remove(i.next());
 return result;
}

public boolean remove(Object k) {
 if (contains(k)) {
  int i = list.size();
  while(i-- != 0) if (comparator().compare(list.get(i), k) == 0) {
   list.remove(i);
   break;
  }
 }
 return super.remove(k);
}

private void justRemove(Object k) { super.remove(k); }

public java.util.Iterator iterator() {
 return new java.util.Iterator() {
   final java.util.Iterator iterator = list.iterator();
   Object curr;
   public Object next() { return curr = iterator.next(); }
   public boolean hasNext() { return iterator.hasNext(); }
   public void remove() {
    justRemove(curr);
    iterator.remove();
   }
  };
 }
}

private
static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

private static String format(double d) {
 StringBuffer s = new StringBuffer();
 return format.format(d, s, fp).toString();
}

private static void speedTest(int n, float f, boolean comp) {
 int i, j;
 OPEN_HASH_BIG_SET m;
 java.util.HashSet t;

 KEY_TYPE k[] = new KEY_TYPE[n];
 KEY_TYPE nk[] = new KEY_TYPE[n];
 long ms;
```

```
for(i = 0; i < n; i++) {
 k[i] = genKey();
 nk[i] = genKey();
}

double totAdd = 0, totYes = 0, totNo = 0, totIter = 0, totRemYes = 0, totRemNo = 0, d;

if (comp) { for(j = 0; j < 20; j++) {

 t = new java.util.HashSet(16);

 /* We add pairs to t. */
 ms = System.currentTimeMillis();
 for(i = 0; i < n;  i++) t.add(KEY2OBJ(k[i]));
 d = 1.0 * n / (System.currentTimeMillis() - ms);
 if (j > 2) totAdd += d;
 System.out.print("Add: " + format(d) +" K/s ");

 /*
We check for pairs in t. */
 ms = System.currentTimeMillis();
 for(i = 0; i < n;  i++) t.contains(KEY2OBJ(k[i]));
 d = 1.0 * n / (System.currentTimeMillis() - ms);
 if (j > 2) totYes += d;
 System.out.print("Yes: " + format(d) +" K/s ");

 /* We check for pairs not in t. */
 ms = System.currentTimeMillis();
 for(i = 0; i < n;  i++) t.contains(KEY2OBJ(nk[i]));
 d = 1.0 * n / (System.currentTimeMillis() - ms);
 if (j > 2) totNo += d;
 System.out.print("No: " + format(d) +" K/s ");

 /* We iterate on t. */
 ms = System.currentTimeMillis();
 for(java.util.Iterator it = t.iterator(); it.hasNext(); it.next());
 d = 1.0 * n / (System.currentTimeMillis() - ms);
 if (j > 2) totIter += d;
 System.out.print("Iter: " + format(d) +" K/s ");

 /* We delete pairs not in t. */
 ms = System.currentTimeMillis();
 for(i = 0; i < n;  i++) t.remove(KEY2OBJ(nk[i]));
 d = 1.0 * n / (System.currentTimeMillis() - ms);
 if (j > 2) totRemNo += d;
 System.out.print("RemNo:
```

```
" + format(d) +" K/s ");

  /* We delete pairs in t. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) t.remove(KEY2OBJ(k[i]));
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totRemYes += d;
  System.out.print("RemYes: " + format(d) +" K/s ");

  System.out.println();
  }

  System.out.println();
  System.out.println("java.util Add: " + format(totAdd/(j-3)) + " K/s Yes: " + format(totYes/(j-3)) + " K/s No: " +
format(totNo/(j-3)) + " K/s Iter: " + format(totIter/(j-3)) + " K/s RemNo: " + format(totRemNo/(j-3)) + " K/s
RemYes: " + format(totRemYes/(j-3)) + "K/s");

  System.out.println();

  totAdd = totYes = totNo = totIter = totRemYes = totRemNo = 0;
  }

  for(j = 0; j < 20; j++) {

  m = new OPEN_HASH_BIG_SET(16, f);

  /* We add pairs to m. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) m.add(k[i]);
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totAdd += d;
  System.out.print("Add: " + format(d)
+" K/s ");

  /* We check for pairs in m. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) m.contains(k[i]);
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totYes += d;
  System.out.print("Yes: " + format(d) +" K/s ");

  /* We check for pairs not in m. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) m.contains(nk[i]);
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totNo += d;
  System.out.print("No: " + format(d) +" K/s ");
```

```
  /* We iterate on m. */
  ms = System.currentTimeMillis();
  for(KEY_ITERATOR it = (KEY_ITERATOR)m.iterator(); it.hasNext(); it.NEXT_KEY());
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totIter += d;
  System.out.print("Iter: " + format(d) +" K/s ");

  /* We delete pairs not in m. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) m.remove(nk[i]);
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totRemNo += d;
  System.out.print("RemNo:
" + format(d) +" K/s ");

  /* We delete pairs in m. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) m.remove(k[i]);
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totRemYes += d;
  System.out.print("RemYes: " + format(d) +" K/s ");

  System.out.println();
  }


 System.out.println();
 System.out.println("fastutil  Add: " + format(totAdd/(j-3)) + " K/s Yes: " + format(totYes/(j-3)) + " K/s No: " +
format(totNo/(j-3)) + " K/s Iter: " + format(totIter/(j-3)) + " K/s RemNo: " + format(totRemNo/(j-3)) + " K/s
RemYes: " + format(totRemYes/(j-3)) + " K/s");

 System.out.println();
 }


 private static void fatal(String msg) {
  System.out.println(msg);
  System.exit(1);
 }

 private static void ensure(boolean cond, String msg) {
  if (cond) return;
  fatal(msg);
 }


 private static void printProbes(OPEN_HASH_BIG_SET m) {
  long totProbes = 0;
```

```
 double totSquareProbes = 0;
 int maxProbes = 0;
 final double f = (double)m.size / m.n;
 for(int
i = 0, c = 0; i < m.n; i++) {
  if (! KEY_IS_NULL(BigArrays.get(m.key, i))) c++;
  else {
   if (c != 0) {
    final long p = (c + 1) * (c + 2) / 2;
    totProbes += p;
    totSquareProbes += (double)p * p;
   }
   maxProbes = Math.max(c, maxProbes);
   c = 0;
   totProbes++;
   totSquareProbes++;
  }
 }

 final double expected = (double)totProbes / m.n;
 System.err.println("Expected probes: " + (
  3 * Math.sqrt(3) * (f / ((1 - f) * (1 - f))) + 4 / (9 * f) - 1
 ) + "; actual: " + expected + "; stddev: " + Math.sqrt(totSquareProbes / m.n - expected * expected)  + "; max probes:
" + maxProbes);
 }
 private static void runTest(int n, float f) {
 int c;
 OPEN_HASH_BIG_SET m = new OPEN_HASH_BIG_SET(Hash.DEFAULT_INITIAL_SIZE, f);
 java.util.Set t = new java.util.HashSet();

 /* First of all, we fill t with random data. */

 for(int i=0; i<f * n;  i++) t.add(KEY2OBJ(genKey()));

 /* Now we add to m the same data */

 m.addAll(t);

 if (!m.equals(t)) System.out.println("Error
(" + seed + "): !m.equals(t) after insertion");
 if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after insertion");
 printProbes(m);

 /* Now we check that m actually holds that data. */

 for(java.util.Iterator i=t.iterator(); i.hasNext();) {
  Object e = i.next();
  if (!m.contains(e)) {
```

```
   System.out.println("Error (" + seed + "): m and t differ on a key ("+e+") after insertion (iterating on t)");
   System.exit(1);
  }
 }

 /* Now we check that m actually holds that data, but iterating on m. */

 c = 0;
 for(java.util.Iterator i=m.iterator(); i.hasNext();) {
  Object e = i.next();
  c++;
  if (!t.contains(e)) {
   System.out.println("Error (" + seed + "): m and t differ on a key ("+e+") after insertion (iterating on m)");
   System.exit(1);
  }
 }

 if (c != t.size()) {
  System.out.println("Error (" + seed + "): m has only " + c + " keys instead of " + t.size() + " after insertion (iterating
on m)");
  System.exit(1);
 }
 /*
 Now we check that inquiries about random data give the same answer in m and t. For
    m we use the polymorphic method. */

 for(int i=0; i<n;  i++) {
  KEY_TYPE T = genKey();
  if (m.contains(T) != t.contains(KEY2OBJ(T))) {
   System.out.println("Error (" + seed + "): divergence in keys between t and m (polymorphic method)");
   System.exit(1);
  }
 }

 /* Again, we check that inquiries about random data give the same answer in m and t, but
    for m we use the standard method. */

 for(int i=0; i<n;  i++) {
  KEY_TYPE T = genKey();
  if (m.contains(KEY2OBJ(T)) != t.contains(KEY2OBJ(T))) {
   System.out.println("Error (" + seed + "): divergence between t and m (standard method)");
   System.exit(1);
  }
 }


 /* Now we put and remove random data in m and t, checking that the result is the same. */
```

```
 for(int i=0; i<20*n;  i++) {
  KEY_TYPE T = genKey();
  if (m.add(KEY2OBJ(T)) != t.add(KEY2OBJ(T))) {
   System.out.println("Error (" + seed + "): divergence
in add() between t and m");
   System.exit(1);
  }
  T = genKey();
  if (m.remove(KEY2OBJ(T)) != t.remove(KEY2OBJ(T))) {
   System.out.println("Error (" + seed + "): divergence in remove() between t and m");
   System.exit(1);
  }
 }

 if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after removal");
 if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after removal");

 /* Now we check that m actually holds that data. */

 for(java.util.Iterator i=t.iterator(); i.hasNext();) {
  Object e = i.next();
  if (!m.contains(e)) {
   System.out.println("Error (" + seed + "): m and t differ on a key ("+e+") after removal (iterating on t)");
   System.exit(1);
  }
 }

 /* Now we check that m actually holds that data, but iterating on m. */

 for(java.util.Iterator i=m.iterator(); i.hasNext();) {
  Object e = i.next();
  if (!t.contains(e)) {
   System.out.println("Error (" + seed + "): m and t differ on a key ("+e+") after
removal (iterating on m)");
   System.exit(1);
  }
 }

 printProbes(m);

 /* Now we make m into an array, make it again a set and check it is OK. */
 KEY_TYPE a[] = m.TO_KEY_ARRAY();

 if (!new OPEN_HASH_BIG_SET(a).equals(m))
  System.out.println("Error (" + seed + "): toArray() output (or array-based constructor) is not OK");

 /* Now we check cloning. */
```

```
  ensure(m.equals(((OPEN_HASH_BIG_SET)m).clone()), "Error (" + seed + "): m does not equal m.clone()");
  ensure(((OPEN_HASH_BIG_SET)m).clone().equals(m), "Error (" + seed + "): m.clone() does not equal m");

  int h = m.hashCode();

  /* Now we save and read m. */

  try {
   java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
   java.io.OutputStream os = new java.io.FileOutputStream(ff);
   java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

   oos.writeObject(m);
   oos.close();

   java.io.InputStream is = new java.io.FileInputStream(ff);
   java.io.ObjectInputStream
 ois = new java.io.ObjectInputStream(is);

   m = (OPEN_HASH_BIG_SET)ois.readObject();
   ois.close();
   ff.delete();
  }
  catch(Exception e) {
   e.printStackTrace();
   System.exit(1);
  }

#if !KEY_CLASS_Reference
  if (m.hashCode() != h) System.out.println("Error (" + seed + "): hashCode() changed after save/read");

  printProbes(m);

  /* Now we check that m actually holds that data, but iterating on m. */

  for(java.util.Iterator i=m.iterator(); i.hasNext();) {
   Object e = i.next();
   if (!t.contains(e)) {
    System.out.println("Error (" + seed + "): m and t differ on a key ("+e+") after save/read");
    System.exit(1);
   }
  }
#else
  m.clear();
  m.addAll(t);
#endif

  /* Now we put and remove random data in m and t, checking that the result is the same. */
```

```
 for(int i=0; i<20*n;  i++) {
  KEY_TYPE T = genKey();
  if (m.add(KEY2OBJ(T)) != t.add(KEY2OBJ(T))) {
   System.out.println("Error (" + seed + "): divergence in add() between t and m after
save/read");
   System.exit(1);
  }
  T = genKey();
  if (m.remove(KEY2OBJ(T)) != t.remove(KEY2OBJ(T))) {
   System.out.println("Error (" + seed + "): divergence in remove() between t and m after save/read");
   System.exit(1);
  }
 }

 if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after post-save/read removal");
 if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after post-save/read removal");


 /* Now we take out of m everything, and check that it is empty. */

 for(java.util.Iterator i=m.iterator(); i.hasNext();) { i.next(); i.remove();}

 if (!m.isEmpty())  {
  System.out.println("Error (" + seed + "): m is not empty (as it should be)");
  System.exit(1);
 }

#if KEY_CLASS_Integer || KEY_CLASS_Long
 m = new OPEN_HASH_BIG_SET(n, f);
 t.clear();
 int x;

 /* Now we torture-test the hash table. This part is implemented only for integers and longs. */

 int p = m.key.length - 1;

 for(int i=0; i<p; i++) {
  for (int
j=0; j<20; j++) {
   m.add(i+(r.nextInt() % 10)*p);
   m.remove(i+(r.nextInt() % 10)*p);
  }

  for (int j=-10; j<10; j++) m.remove(i+j*p);
 }

 t.addAll(m);
```

```
  /* Now all table entries are REMOVED. */

  int k = 0;
  for(int i=0; i<(p*f)/10; i++) {
   for (int j=0; j<10; j++) {
    k++;
    x = i+(r.nextInt() % 10)*p;
    if (m.add(x) != t.add(KEY2OBJ(x)))
     System.out.println("Error (" + seed + "): m and t differ on a key during torture-test insertion.");
   }
  }

  if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after torture-test insertion");
  if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after torture-test insertion");

  for(int i=0; i<(p*f)/10; i++) {
   for (int j=0; j<10; j++) {
    x = i+(r.nextInt() % 10)*p;
    if (m.remove(x) != t.remove(KEY2OBJ(x)))
     System.out.println("Error (" + seed + "): m and t differ on a key during torture-test removal.");
   }
  }

  if (!m.equals(t)) System.out.println("Error
 (" + seed + "): !m.equals(t) after torture-test removal");
  if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after torture-test removal");

  if (!m.equals(m.clone())) System.out.println("Error (" + seed + "): !m.equals(m.clone()) after torture-test removal");
  if (!((OPEN_HASH_BIG_SET)m.clone()).equals(m)) System.out.println("Error (" + seed + "): !m.clone().equals(m)
after torture-test removal");

  m.trim();

  if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after trim()");
  if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after trim()");

#endif

  System.out.println("Test OK");
  return;
 }


 public static void main(String args[]) {
  float f = Hash.DEFAULT_LOAD_FACTOR;
  int n  = Integer.parseInt(args[1]);
  if (args.length>2) f = Float.parseFloat(args[2]);
```

```
  if (args.length > 3) r = new java.util.Random(seed = Long.parseLong(args[3]));

  try {
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0]))
 speedTest(n, f, "speedComp".equals(args[0]));
   else if ("test".equals(args[0])) runTest(n, f);
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed: " + seed);
  }
 }

#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/OpenHashBigSet.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.List;
import java.util.Collection;
import java.util.Random;
import java.util.RandomAccess;

/** A class providing static methods and objects that do useful things with type-specific lists.
*
* @see java.util.Collections
```

```
*/

public final class LISTS {

 private LISTS() {}

 /** Shuffles
 the specified list using the specified pseudorandom number generator.
  *
  * @param l the list to be shuffled.
  * @param random a pseudorandom number generator.
  * @return {@code l}.
  */
 public static KEY_GENERIC LIST KEY_GENERIC shuffle(final LIST KEY_GENERIC l, final Random random)
{
  for(int i = l.size(); i-- != 0;) {
   final int p = random.nextInt(i + 1);
   final KEY_GENERIC_TYPE t = l.GET_KEY(i);
   l.set(i, l.GET_KEY(p));
   l.set(p, t);
  }
  return l;
 }

 /** An immutable class representing an empty type-specific list.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific list.
  */

 public static class EmptyList KEY_GENERIC extends COLLECTIONS.EmptyCollection KEY_GENERIC
implements LIST KEY_GENERIC, RandomAccess, java.io.Serializable, Cloneable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected EmptyList() {}

  @Override
  public KEY_GENERIC_TYPE GET_KEY(int i) { throw new IndexOutOfBoundsException();
  }

  @Override
  public boolean REMOVE(KEY_TYPE k) { throw new UnsupportedOperationException(); }

  @Override
  public KEY_GENERIC_TYPE REMOVE_KEY(int i) { throw new UnsupportedOperationException(); }

  @Override
```

```
  public void add(final int index, final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException(); }

  @Override
  public KEY_GENERIC_TYPE set(final int index, final KEY_GENERIC_TYPE k) { throw new
UnsupportedOperationException(); }

  @Override
  public int indexOf(KEY_TYPE k) { return -1; }

  @Override
  public int lastIndexOf(KEY_TYPE k) { return -1; }

  @Override
  public boolean addAll(int i, Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
  @Override
  public boolean addAll(LIST c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(int i, COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(int i, LIST
 c) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @SuppressWarnings("deprecation")
  @Deprecated
  @Override
  public void add(final int index, final KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @SuppressWarnings("deprecation")
  @Deprecated
  @Override
  public KEY_CLASS get(final int index) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @SuppressWarnings("deprecation")
  @Deprecated
  @Override
  public boolean add(final KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException(); }
```

```
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @SuppressWarnings("deprecation")
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS
 set(final int index, final KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @SuppressWarnings("deprecation")
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS remove(int k) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @SuppressWarnings("deprecation")
  @Deprecated
  @Override
  public int indexOf(Object k) { return -1; }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @SuppressWarnings("deprecation")
  @Deprecated
  @Override
  public int lastIndexOf(Object k) { return -1; }

  // Empty lists are trivially always sorted
  @Override
  public void sort(final KEY_COMPARATOR comparator) { }

  @Override
  public void unstableSort(final KEY_COMPARATOR comparator) { }
#endif

  // Empty lists are trivially
 always sorted
  DEPRECATED_IF_KEYS_PRIMITIVE
  @Override
  public void sort(final java.util.Comparator<? super KEY_GENERIC_CLASS> comparator) { }

  DEPRECATED_IF_KEYS_PRIMITIVE
  @Override
  public void unstableSort(final java.util.Comparator<? super KEY_GENERIC_CLASS> comparator) { }

  SUPPRESS_WARNINGS_KEY_UNCHECKED
```

```
  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC listIterator() { return ITERATORS.EMPTY_ITERATOR; }

  SUPPRESS_WARNINGS_KEY_UNCHECKED
  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC iterator() { return ITERATORS.EMPTY_ITERATOR; }

  SUPPRESS_WARNINGS_KEY_UNCHECKED
  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC listIterator(int i) { if (i == 0) return
ITERATORS.EMPTY_ITERATOR; throw new IndexOutOfBoundsException(String.valueOf(i)); }

  @Override
  public LIST KEY_GENERIC subList(int from, int to) { if (from == 0 && to == 0) return this; throw new
IndexOutOfBoundsException(); }

  @Override
  public void getElements(int from, KEY_TYPE[] a, int offset, int
 length) { if (from == 0 && length == 0 && offset >= 0 && offset <= a.length) return; throw new
IndexOutOfBoundsException(); }

  @Override
  public void removeElements(int from, int to) { throw new UnsupportedOperationException(); }

  @Override
  public void addElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) { throw new
UnsupportedOperationException(); }

  @Override
  public void addElements(int index, final KEY_GENERIC_TYPE a[]) { throw new
UnsupportedOperationException(); }

  @Override
  public void setElements(final KEY_GENERIC_TYPE a[]) { throw new UnsupportedOperationException(); }

  @Override
  public void setElements(int index, final KEY_GENERIC_TYPE a[]) { throw new
UnsupportedOperationException(); }

  @Override
  public void setElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) { throw new
UnsupportedOperationException(); }

  @Override
  public void size(int s)  { throw new UnsupportedOperationException(); }

#if ! KEY_CLASS_Reference
  @Override
```

```
  public
 int compareTo(final List<? extends KEY_GENERIC_CLASS> o) {
   if (o == this) return 0;
   return ((List<?>)o).isEmpty() ? 0 : -1;
  }
#endif

  @Override
  public Object clone() { return EMPTY_LIST; }

  @Override
  public int hashCode() { return 1; }

  @Override
  @SuppressWarnings("rawtypes")
  public boolean equals(Object o) { return o instanceof List && ((List)o).isEmpty(); }

  @Override
  public String toString() { return "[]"; }

  private Object readResolve() { return EMPTY_LIST; }
 }

 /** An empty list (immutable). It is serializable and cloneable.
  */
 SUPPRESS_WARNINGS_KEY_RAWTYPES
 public static final EmptyList EMPTY_LIST = new EmptyList();

#if KEYS_REFERENCE
 /** Returns an empty list (immutable). It is serializable and cloneable.
  *
  * <p>This method provides a typesafe access to {@link #EMPTY_LIST}.
  * @return an empty list (immutable).
  */
 @SuppressWarnings("unchecked")
 public static KEY_GENERIC LIST KEY_GENERIC emptyList() {
  return
 EMPTY_LIST;
 }
#endif


 /** An immutable class representing a type-specific singleton list.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific list.
  */
```

```
 public static class Singleton KEY_GENERIC extends ABSTRACT_LIST KEY_GENERIC implements
RandomAccess, java.io.Serializable, Cloneable {

  private static final long serialVersionUID = -7046029254386353129L;

  private final KEY_GENERIC_TYPE element;

  protected Singleton(final KEY_GENERIC_TYPE element) { this.element = element; }

  @Override
  public KEY_GENERIC_TYPE GET_KEY(final int i) { if (i == 0) return element; throw new
IndexOutOfBoundsException(); }

  @Override
  public boolean REMOVE(KEY_TYPE k) { throw new UnsupportedOperationException(); }

  @Override
  public KEY_GENERIC_TYPE REMOVE_KEY(final int i) { throw new UnsupportedOperationException(); }

  @Override
  public boolean contains(final KEY_TYPE k) { return KEY_EQUALS(k, element); }

  /* Slightly optimized w.r.t.
 the one in ABSTRACT_SET. */

  @Override
  public KEY_TYPE[] TO_KEY_ARRAY() {
   KEY_TYPE a[] = new KEY_TYPE[1];
   a[0] = element;
   return a;
  }

  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC listIterator() { return ITERATORS.singleton(element); }

  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC iterator() { return listIterator(); }

  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC listIterator(final int i) {
   if (i > 1 || i < 0) throw new  IndexOutOfBoundsException();
   final KEY_LIST_ITERATOR KEY_GENERIC l = listIterator();
   if (i == 1) l.NEXT_KEY();
   return l;
  }

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
```

```
  public LIST KEY_GENERIC subList(final int from, final int to) {
   ensureIndex(from);
   ensureIndex(to);
   if (from > to) throw new IndexOutOfBoundsException("Start index (" + from + ") is greater than end index (" + to +
")");

   if (from != 0 || to != 1) return EMPTY_LIST;
   return this;
  }

  @Override
  public boolean addAll(int i, Collection<?
 extends KEY_GENERIC_CLASS> c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

  @Override
  public boolean removeAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean retainAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE

  @Override
  public boolean addAll(LIST c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(int i, LIST c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(int i, COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean removeAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean retainAll(final
 COLLECTION c) { throw new UnsupportedOperationException(); }

  // Lists of size 1 are trivially always sorted
  @Override
  public void sort(final KEY_COMPARATOR comparator) { }
```

```
  @Override
  public void unstableSort(final KEY_COMPARATOR comparator) { }

#endif

  // Lists of size 1 are trivially always sorted
  DEPRECATED_IF_KEYS_PRIMITIVE
  @Override
  public void sort(final java.util.Comparator<? super KEY_GENERIC_CLASS> comparator) { }

  DEPRECATED_IF_KEYS_PRIMITIVE
  @Override
  public void unstableSort(final java.util.Comparator<? super KEY_GENERIC_CLASS> comparator) { }

  @Override
  public void removeElements(int from, int to) { throw new UnsupportedOperationException(); }

  @Override
  public void addElements(int index, KEY_GENERIC_TYPE a[]) { throw new UnsupportedOperationException(); }

  @Override
  public void addElements(int index, KEY_GENERIC_TYPE a[], int offset, int length) { throw new
UnsupportedOperationException(); }

  @Override
  public void setElements(final
 KEY_GENERIC_TYPE a[]) { throw new UnsupportedOperationException(); }

  @Override
  public void setElements(int index, final KEY_GENERIC_TYPE a[]) { throw new
UnsupportedOperationException(); }

  @Override
  public void setElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) { throw new
UnsupportedOperationException(); }

  @Override
  public int size() { return 1; }

  @Override
  public void size(final int size) { throw new UnsupportedOperationException(); }

  @Override
  public void clear() { throw new UnsupportedOperationException(); }

  @Override
  public Object clone() { return this; }
```

```
 }

 /** Returns a type-specific immutable list containing only the specified element. The returned list is serializable and cloneable.
  *
  * @param element the only element of the returned list.
  * @return a type-specific immutable list containing just {@code element}.
  */

 public static KEY_GENERIC LIST KEY_GENERIC singleton(final KEY_GENERIC_TYPE element) { return new Singleton KEY_GENERIC_DIAMOND(element); }

#if KEYS_PRIMITIVE

 /** Returns a type-specific immutable list containing only the specified element. The returned list is serializable and cloneable.
  *
  * @param element the only element of the returned list.
  * @return a type-specific immutable list containing just {@code element}.
  */

 public static KEY_GENERIC LIST KEY_GENERIC singleton(final Object element) { return new Singleton KEY_GENERIC_DIAMOND(KEY_OBJ2TYPE(element)); }

#endif


 /** A synchronized wrapper class for lists. */

 public static class SynchronizedList KEY_GENERIC extends COLLECTIONS.SynchronizedCollection KEY_GENERIC implements LIST KEY_GENERIC, java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final LIST KEY_GENERIC list; // Due to the large number of methods that are not in COLLECTION, this is worth caching.

  protected SynchronizedList(final LIST KEY_GENERIC l, final Object sync) {
   super(l,
 sync);
   this.list = l;
  }

  protected SynchronizedList(final LIST KEY_GENERIC l) {
   super(l);
   this.list = l;
  }
```

```
  @Override
  public KEY_GENERIC_TYPE GET_KEY(final int i) { synchronized(sync) { return list.GET_KEY(i); } }

  @Override
  public KEY_GENERIC_TYPE set(final int i, final KEY_GENERIC_TYPE k) { synchronized(sync) { return
list.set(i, k); } }

  @Override
  public void add(final int i, final KEY_GENERIC_TYPE k) { synchronized(sync) { list.add(i, k); } }

  @Override
  public KEY_GENERIC_TYPE REMOVE_KEY(final int i) { synchronized(sync) { return list.REMOVE_KEY(i); }
}

  @Override
  public int indexOf(final KEY_TYPE k) { synchronized(sync) { return list.indexOf(k); } }

  @Override
  public int lastIndexOf(final KEY_TYPE k) { synchronized(sync) { return list.lastIndexOf(k); } }

  @Override
  public boolean addAll(final int index, final Collection<? extends KEY_GENERIC_CLASS> c) {
synchronized(sync) { return list.addAll(index, c); } }

  @Override
  public void getElements(final
 int from, final KEY_TYPE a[], final int offset, final int length) { synchronized(sync) { list.getElements(from, a,
offset, length); } }

  @Override
  public void removeElements(final int from, final int to) { synchronized(sync) { list.removeElements(from, to); } }

  @Override
  public void addElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) { synchronized(sync) {
list.addElements(index, a, offset, length); } }

  @Override
  public void addElements(int index, final KEY_GENERIC_TYPE a[]) { synchronized(sync) {
list.addElements(index, a); } }

  @Override
  public void setElements(final KEY_GENERIC_TYPE a[]) { synchronized(sync) { list.setElements(a); } }

  @Override
  public void setElements(int index, final KEY_GENERIC_TYPE a[]) { synchronized(sync) { list.setElements(index,
a); } }

  @Override
```

```
  public void setElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) { synchronized(sync) {
list.setElements(index, a, offset, length);
 } }

  @Override
  public void size(final int size) { synchronized(sync) { list.size(size); } }

  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC listIterator() { return list.listIterator(); }

  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC iterator() { return listIterator(); }

  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC listIterator(final int i) { return list.listIterator(i); }

  @Override
  public LIST KEY_GENERIC subList(final int from, final int to) { synchronized(sync) { return new
SynchronizedList KEY_GENERIC_DIAMOND(list.subList(from, to), sync); } }

  @Override
  public boolean equals(final Object o) { if (o == this) return true; synchronized(sync) { return collection.equals(o); }
}

  @Override
  public int hashCode() { synchronized(sync) { return collection.hashCode(); } }

#if ! KEY_CLASS_Reference
  @Override
  public int compareTo(final List<? extends KEY_GENERIC_CLASS> o) { synchronized(sync) { return
list.compareTo(o); } }
#endif

#if KEYS_PRIMITIVE
  @Override
  public
 boolean addAll(final int index, final COLLECTION c) { synchronized(sync) { return list.addAll(index, c); } }

  @Override
  public boolean addAll(final int index, LIST l) { synchronized(sync) { return list.addAll(index, l); } }

  @Override
  public boolean addAll(LIST l) { synchronized(sync) { return list.addAll(l); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
```

```
  public KEY_GENERIC_CLASS get(final int i) { synchronized(sync) { return list.get(i); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public void add(final int i, KEY_GENERIC_CLASS k) { synchronized(sync) { list.add(i, k); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS set(final int index, KEY_GENERIC_CLASS k) { synchronized(sync) { return
 list.set(index, k); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS remove(final int i) { synchronized(sync) { return list.remove(i); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public int indexOf(final Object o) { synchronized(sync) { return list.indexOf(o); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public int lastIndexOf(final Object o) { synchronized(sync) { return list.lastIndexOf(o); } }

  @Override
  public void sort(final KEY_COMPARATOR comparator) { synchronized(sync) { list.sort(comparator); } }

  @Override
  public void unstableSort(final KEY_COMPARATOR comparator) { synchronized(sync) {
list.unstableSort(comparator); } }
#endif

  DEPRECATED_IF_KEYS_PRIMITIVE
  @Override
  public void sort(final
 java.util.Comparator<? super KEY_GENERIC_CLASS> comparator) { synchronized(sync) {list.sort(comparator);
} }

  DEPRECATED_IF_KEYS_PRIMITIVE
```

```
  @Override
  public void unstableSort(final java.util.Comparator<? super KEY_GENERIC_CLASS> comparator) {
synchronized(sync) {list.unstableSort(comparator); } }

  private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
   synchronized(sync) { s.defaultWriteObject(); }
  }
 }


 /** A synchronized wrapper class for random-access lists. */

 public static class SynchronizedRandomAccessList KEY_GENERIC extends SynchronizedList KEY_GENERIC
implements RandomAccess, java.io.Serializable {
  private static final long serialVersionUID = 0L;

  protected SynchronizedRandomAccessList(final LIST KEY_GENERIC l, final Object sync) {
   super(l, sync);
  }

  protected SynchronizedRandomAccessList(final LIST KEY_GENERIC l) {
   super(l);
  }

  @Override
  public LIST KEY_GENERIC subList(final int from, final int to) { synchronized(sync)
 { return new SynchronizedRandomAccessList KEY_GENERIC_DIAMOND(list.subList(from, to), sync); } }
 }


 /** Returns a synchronized type-specific list backed by the given type-specific list.
  *
  * @param l the list to be wrapped in a synchronized list.
  * @return a synchronized view of the specified list.
  * @see java.util.Collections#synchronizedList(List)
  */
 public static KEY_GENERIC LIST KEY_GENERIC synchronize(final LIST KEY_GENERIC l) {
  return l instanceof RandomAccess ? new SynchronizedRandomAccessList KEY_GENERIC_DIAMOND(l) : new
SynchronizedList KEY_GENERIC_DIAMOND(l);
 }

 /** Returns a synchronized type-specific list backed by the given type-specific list, using an assigned object to
synchronize.
  *
  * @param l the list to be wrapped in a synchronized list.
  * @param sync an object that will be used to synchronize the access to the list.
  * @return a synchronized view of the specified list.
  * @see java.util.Collections#synchronizedList(List)
```

```
 */

 public
 static KEY_GENERIC LIST KEY_GENERIC synchronize(final LIST KEY_GENERIC l, final Object sync) {
  return l instanceof RandomAccess ? new SynchronizedRandomAccessList KEY_GENERIC_DIAMOND(l, sync) :
new SynchronizedList KEY_GENERIC_DIAMOND(l, sync);
 }



 /** An unmodifiable wrapper class for lists. */

 public static class UnmodifiableList KEY_GENERIC extends COLLECTIONS.UnmodifiableCollection
KEY_GENERIC implements LIST KEY_GENERIC, java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final LIST KEY_GENERIC list; // Due to the large number of methods that are not in COLLECTION,
this is worth caching.

  protected UnmodifiableList(final LIST KEY_GENERIC l) {
   super(l);
   this.list = l;
  }

  @Override
  public KEY_GENERIC_TYPE GET_KEY(final int i) { return list.GET_KEY(i); }

  @Override
  public KEY_GENERIC_TYPE set(final int i, final KEY_GENERIC_TYPE k) { throw new
UnsupportedOperationException(); }

  @Override
  public
 void add(final int i, final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException(); }

  @Override
  public KEY_GENERIC_TYPE REMOVE_KEY(final int i) { throw new UnsupportedOperationException(); }

  @Override
  public int indexOf(final KEY_TYPE k) { return list.indexOf(k); }

  @Override
  public int lastIndexOf(final KEY_TYPE k) { return list.lastIndexOf(k); }

  @Override
  public boolean addAll(final int index, final Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }
```

```
  @Override
  public void getElements(final int from, final KEY_TYPE a[], final int offset, final int length) {
list.getElements(from, a, offset, length); }

  @Override
  public void removeElements(final int from, final int to) { throw new UnsupportedOperationException(); }

  @Override
  public void addElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) { throw new
UnsupportedOperationException(); }

  @Override
  public void addElements(int index,
 final KEY_GENERIC_TYPE a[]) { throw new UnsupportedOperationException(); }

  @Override
  public void setElements(final KEY_GENERIC_TYPE a[]) { throw new UnsupportedOperationException(); }

  @Override
  public void setElements(int index, final KEY_GENERIC_TYPE a[]) { throw new
UnsupportedOperationException(); }

  @Override
  public void setElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) { throw new
UnsupportedOperationException(); }

  @Override
  public void size(final int size) { list.size(size); }

  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC listIterator() { return
ITERATORS.unmodifiable(list.listIterator()); }

  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC iterator() { return listIterator(); }

  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC listIterator(final int i) { return
ITERATORS.unmodifiable(list.listIterator(i)); }

  @Override
  public LIST KEY_GENERIC subList(final int from, final int to) { return new UnmodifiableList
KEY_GENERIC_DIAMOND(list.subList(from,
 to)); }

  @Override
  public boolean equals(final Object o) { if (o == this) return true; return collection.equals(o); }
```

```
  @Override
  public int hashCode() { return collection.hashCode(); }

#if ! KEY_CLASS_Reference
  @Override
  public int compareTo(final List<? extends KEY_GENERIC_CLASS> o) { return list.compareTo(o); }
#endif

#if KEYS_PRIMITIVE
  @Override
  public boolean addAll(final int index, final COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(final LIST l) { throw new UnsupportedOperationException(); }

  @Override
  public boolean addAll(final int index, final LIST l) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS get(final int i) { return list.get(i); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead.
 */
  @Deprecated
  @Override
  public void add(final int i, KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS set(final int index, KEY_GENERIC_CLASS k) { throw new
UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS remove(final int i) { throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
```

```
  @Override
  public int indexOf(final Object o) { return list.indexOf(o); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public int lastIndexOf(final Object o) { return list.lastIndexOf(o); }

  @Override
  public
 void sort(final KEY_COMPARATOR comparator) { throw new UnsupportedOperationException(); }

  @Override
  public void unstableSort(final KEY_COMPARATOR comparator) { throw new UnsupportedOperationException();
}
#endif
  DEPRECATED_IF_KEYS_PRIMITIVE
  @Override
  public void sort(final java.util.Comparator<? super KEY_GENERIC_CLASS> comparator) { throw new
UnsupportedOperationException(); }

  DEPRECATED_IF_KEYS_PRIMITIVE
  @Override
  public void unstableSort(final java.util.Comparator<? super KEY_GENERIC_CLASS> comparator) { throw new
UnsupportedOperationException(); }
 }


 /** An unmodifiable wrapper class for random-access lists. */

 public static class UnmodifiableRandomAccessList KEY_GENERIC extends UnmodifiableList KEY_GENERIC
implements RandomAccess, java.io.Serializable {

  private static final long serialVersionUID = 0L;

  protected UnmodifiableRandomAccessList(final LIST KEY_GENERIC l) {
   super(l);
  }


  @Override
  public LIST KEY_GENERIC subList(final int from,
 final int to) { return new UnmodifiableRandomAccessList KEY_GENERIC_DIAMOND(list.subList(from, to)); }
 }


 /** Returns an unmodifiable type-specific list backed by the given type-specific list.
```

```
 *
 * @param l the list to be wrapped in an unmodifiable list.
 * @return an unmodifiable view of the specified list.
 * @see java.util.Collections#unmodifiableList(List)
 */
 public static KEY_GENERIC LIST KEY_GENERIC unmodifiable(final LIST KEY_GENERIC l) {
  return l instanceof RandomAccess ? new UnmodifiableRandomAccessList KEY_GENERIC_DIAMOND(l) : new
UnmodifiableList KEY_GENERIC_DIAMOND(l);
 }




#ifdef TEST

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif KEY_CLASS_Object
  return Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }


 private static void testLists(KEY_TYPE k, LIST m, List t, int level) {
  int
 n = 100;
  int c;

  long ms;
  boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement, mThrowsIndex, tThrowsIndex,
mThrowsUnsupp, tThrowsUnsupp;
  boolean rt = false, rm = false;
  Object Rt = null, Rm = null;

  if (level == 0) return;

  /* Now we check that m and t are equal. */
  if (!m.equals(t) || ! t.equals(m)) System.err.println("m: " + m + " t: " + t);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");

  /* Now we check that m actually holds that data. */
```

```
  for(java.util.Iterator i=t.iterator(); i.hasNext();) {
   ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */
  for(java.util.Iterator i=m.listIterator(); i.hasNext();) {
   ensure(t.contains(i.next()), "Error (" + level +
 ", " + seed + "): m and t differ on an entry after insertion (iterating on m)");
  }

  /* Now we check that inquiries about random data give the same answer in m and t. For
     m we use the polymorphic method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    m.contains(T);
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }

   try {
    t.contains(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence
 in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " +
m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex) ensure(m.contains(KEY2OBJ(T)) ==
t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in keys between t and m (polymorphic
method) " + m);
  }

  /* Again, we check that inquiries about random data give the same answer in m and t, but
     for m we use the standard method. */
```

```
  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement
 = mThrowsIllegal = tThrowsIllegal = mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    m.contains(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    t.contains(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", "
 + seed + "): contains() divergence in IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " +
tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): contains() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp)
ensure(m.contains(KEY2OBJ(T)) ==  t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence
between t and m (standard method) " + m);
  }

  /* Now we add and remove random data in m and t, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal
 = tThrowsIllegal = mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    rm = m.add(KEY2OBJ(T));
   }
```

```
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    rt = t.add(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() divergence
 in IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): add() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): add() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in add() between t and m " + m);

   T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    rm = m.remove(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) {
 mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    rt = t.remove(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   if (! KEY_EQUALS(T, k) && mThrowsUnsupp && ! tThrowsUnsupp) mThrowsUnsupp = true; // Stupid bug in
Collections.singleton()
```

```
   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal
 + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): remove() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): remove() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in remove() between t and m " + m);
  }

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal " + m);
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal " + m);

  /* Now we add and remove random data in m and t at specific positions, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex
 = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal = mThrowsUnsupp =
tThrowsUnsupp = false;

   int pos = r.nextInt(2);

   try {
    m.add(pos, KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    t.add(pos, KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add() at " + pos + "
divergence in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement
+ ") "
```

```
 + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() at " + pos + " divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): add() at " + pos + " divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): add() at " + pos + " divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);

   T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   pos = r.nextInt(2);

   try {
    Rm = m.remove(pos);
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true;
 }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    Rt = t.remove(pos);
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }


   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() at " + pos + "
divergence in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement
+ ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() at " + pos + " divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): remove() at " + pos +
 " divergence in IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): remove() at " + pos + " divergence
in UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(Rm == Rt || Rm
!= null && Rm.equals(Rt), "Error (" + level + ", " + seed + "): divergence in remove() at " + pos + " between t and
m " + m);
  }

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal " + m);
```

```
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal " + m);

  /* Now we add and remove random collections in m and t, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal
 = mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    rm = m.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    rt = t.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): addAll() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal
 == tThrowsIllegal, "Error (" + level + ", " + seed + "): addAll() divergence in IllegalArgumentException for " + T +
" (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): addAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): addAll() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in addAll() between t and m " + m);

   T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    rm = m.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
   }
   catch
```

```
 (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    rt = t.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }


   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): removeAll() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): removeAll() divergence in
IllegalArgumentException for " + T + " ("
 + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): removeAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): removeAll() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in removeAll() between t and m " + m);
   }

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after set removal " + m);
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after set removal " + m);

  /* Now we add random collections at specific positions in m and t, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE
 T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   int pos = r.nextInt(2);

   try {
    rm = m.addAll(pos, java.util.Collections.singleton(KEY2OBJ(T)));
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }
```

```
  try {
   rt = t.addAll(pos, java.util.Collections.singleton(KEY2OBJ(T)));
  }
  catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
  catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): addAll() at
 " + pos + " divergence in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " +
tThrowsNoElement + ") " + m);
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): addAll() at " + pos + " divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
  ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): addAll() at " + pos + " divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
  ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): addAll() at " + pos + " divergence
in UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
  if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in addAll() at " + pos + " between t and m " + m);

 }

 ensure(m.equals(t),
"Error (" + level + ", " + seed + "): ! m.equals(t) after set removal " + m);
 ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after set removal " + m);

 /* Now we check that m actually holds the same data. */

 for(java.util.Iterator i=t.iterator(); i.hasNext();) {
  ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
t)");
 }

 /* Now we check that m actually holds that data, but iterating on m. */

 for(java.util.Iterator i=m.listIterator(); i.hasNext();) {
  ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
m)");
 }

 if (m instanceof Singleton) {
  ensure(m.equals(((Singleton)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
  ensure(((Singleton)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
 }

 int h = m.hashCode();
```

```
 /* Now
we save and read m. */

 LIST m2 = null;

 try {
  java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
  java.io.OutputStream os = new java.io.FileOutputStream(ff);
  java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

  oos.writeObject(m);
  oos.close();

  java.io.InputStream is = new java.io.FileInputStream(ff);
  java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

  m2 = (LIST)ois.readObject();
  ois.close();
  ff.delete();
 }
 catch(Exception e) {
  e.printStackTrace();
  System.exit(1);
 }

#if ! KEY_CLASS_Reference

 ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

 /* Now we check that m2 actually holds that data. */

 ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
 ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
#endif

 if
(! m.isEmpty()) {
  int start = r.nextInt(m.size());
  int end = start + r.nextInt(m.size() - start);
  //System.err.println("Checking subList from " + start + " to " + end + " (level=" + (level+1) + ")...");
  testLists(k, m.subList(start, end), t.subList(start, end), level - 1);

  ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after subList");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subList");

 }

 return;
```

```
}

private static void test() {
 KEY_TYPE k = genKey();
 LIST m = new Singleton(k);
 List u = java.util.Collections.singletonList(KEY2OBJ(k));
 testLists(k, m, java.util.Collections.unmodifiableList(u), 3);
 System.out.println("Test OK");
}

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition fp
= new java.text.FieldPosition(0);

private static String format(double d) {
 StringBuffer s = new StringBuffer();
 return format.format(d, s, fp).toString();
}

private static void fatal(String msg) {
 System.out.println(msg);
 System.exit(1);
}

private static void ensure(boolean cond, String msg) {
 if (cond) return;
 fatal(msg);
}

/** This method expects as first argument a lower-cased type (e.g., "int"),
 * and as second optional argument a seed. */

public static void main(String arg[]) throws Exception {
 if (arg.length > 1) r = new java.util.Random(seed = Long.parseLong(arg[1]));

 try {
  test();
 } catch(Throwable e) {
  e.printStackTrace(System.err);
  System.err.println("seed: " + seed);
 }
}

#endif
```

}

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Lists.drv
No license file was found, but licenses were detected in source scan.


```
<project>
 <modelVersion>4.0.0</modelVersion>
 <groupId>it.unimi.dsi</groupId>
 <artifactId>fastutil</artifactId>
 <packaging>jar</packaging>
 <name>fastutil</name>
 <version>VERSION</version>
 <description>fastutil extends the Java Collections Framework by providing type-specific maps, sets, lists and priority queues with a small memory footprint and fast access and insertion; provides also big (64-bit) arrays, sets and lists, and fast, practical I/O classes for binary and text files.</description>
 <url>http://fastutil.di.unimi.it/</url>
 <licenses>
   <license>
     <name>Apache License, Version 2.0</name>
     <url>http://www.apache.org/licenses/LICENSE-2.0.html</url>
     <distribution>repo</distribution>
   </license>
 </licenses>
 <scm>
   <connection>scm:git://github.com/vigna/fastutil.git</connection>
   <url>https://github.com/vigna/fastutil</url>
 </scm>
 <developers>
   <developer>

     <id>vigna</id>
     <name>Sebastiano Vigna</name>
     <email>vigna@di.unimi.it</email>
   </developer>
 </developers>
 <properties>
   <maven.compiler.source>1.8</maven.compiler.source>
   <maven.compiler.target>1.8</maven.compiler.target>
 </properties>
 <dependencies>
   <dependency>
     <groupId>junit</groupId>
     <artifactId>junit</artifactId>
     <version>4.12</version>
     <scope>test</scope>
   </dependency>
```

```
 </dependencies>
</project>
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/pom-model.xml
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2017-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/HashCommonTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ArraysTest.java
*
 /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericOpenHashTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ObjectBigArrayBigListTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/io/FastMultiByteArrayInputStreamTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ObjectAVLTreeSetTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntSemiIndirectHeapsTest.java
*
 /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/Reference2ReferenceArrayMapTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/Object2IntOpenHashMapTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-

gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntListsTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntArrayIndirectPriorityQueueTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2ObjectFunctionTest.java
*
/opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntArrayFIFOQueueTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/chars/CharArrayFrontCodedListTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2IntMapsTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntArraysTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ReferenceArraySetTest.java
*
/opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ObjectArraySetTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/AbstractInt2IntMapTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/io/InspectableFileCachedInputStreamTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/io/BinIOTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntHeapSemiIndirectPriorityQueueTest.java
*
/opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/chars/CharArraysTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/StripedInt2IntOpenHashMapTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/AbstractIntCollectionTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/io/FastByteArrayOutputStreamTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ObjectArrayListTest.java
*
/opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntBigArraysTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ObjectRBTreeSetTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericDefaultTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-

gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/floats/FloatOpenHashSetTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericRBTest.java
*
 /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntOpenHashSetTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/doubles/DoubleArraysTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/AbstractObject2IntFunctionTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/shorts/ShortArraysTest.java
*
 /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/longs/LongOpenHashBigSetTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericOpenHashTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/io/TextIOTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ObjectArrayPriorityQueueTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericAVLTest.java
*
 /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntArrayPriorityQueueTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/io/FastBufferedInputStreamTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericAVLTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2IntOpenHashMapTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/bytes/ByteArrayFrontCodedListTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/doubles/DoubleBigArraysTest.java
*
 /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ObjectHeapPriorityQueueTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/bytes/ByteArraysTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntSetsTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntCollectionsTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-

gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericArrayTest.java
*
/opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/shorts/ShortArrayFrontCodedListTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ObjectBigArraysTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ObjectOpenHashSetTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/doubles/DoubleOpenHashSetTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntBigArrayBigListTest.java
*
/opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/longs/LongArrayFrontCodedListTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/BigArraysTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericDefaultTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ObjectArraysTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericRBTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntArraySetTest.java
*
/opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ObjectSetsTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/SafeMath.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ObjectOpenHashBigSetTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/Object2ObjectArrayMapTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntArrayFrontCodedListTest.java
*
/opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/io/FastBufferedOutputStreamTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericArrayTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntArrayListTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/longs/LongArraysTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-
gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/floats/FloatArraysTest.java
*

/opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntHeapPriorityQueueTest.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2010-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*
* For the sorting code:
*
* Copyright (C) 1999 CERN - European Organization for Nuclear Research.
*
*   Permission to use, copy, modify, distribute and sell this software and
*   its documentation for any purpose is hereby granted without fee,
*   provided that the above copyright notice appear in all copies and that
 *   both that copyright notice and this permission notice appear in
*   supporting documentation. CERN makes no representations about the
*   suitability of this software for any purpose. It is provided "as is"
*   without expressed or implied warranty.
*/
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/BigArraysCommon.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import static it.unimi.dsi.fastutil.BigArrays.grow;
import static it.unimi.dsi.fastutil.BigArrays.length;
import static it.unimi.dsi.fastutil.BigArrays.set;
import static it.unimi.dsi.fastutil.BigArrays.trim;

import java.util.Iterator;
import java.util.ListIterator;
import java.util.NoSuchElementException;
import
 java.util.Objects;

/** A class providing static methods and objects that do useful things with type-specific iterators.
*
* @see Iterator
*/

public final class ITERATORS {

 private ITERATORS() {}

 /** A class returning no elements and a type-specific iterator interface.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific iterator.
  */

 public static class EmptyIterator KEY_GENERIC implements KEY_LIST_ITERATOR KEY_GENERIC,
java.io.Serializable, Cloneable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected EmptyIterator() {}

  @Override
  public boolean hasNext() { return false; }

  @Override
  public boolean hasPrevious() { return false; }

  @Override
```

```
 public KEY_GENERIC_TYPE NEXT_KEY() { throw new NoSuchElementException(); }

 @Override
 public KEY_GENERIC_TYPE PREV_KEY() { throw new NoSuchElementException(); }

 @Override
 public int nextIndex() { return 0; }

 @Override
 public
int previousIndex() { return -1; }

 @Override
 public int skip(int n) { return 0; };

 @Override
 public int back(int n) { return 0; };

 @Override
 public Object clone() { return EMPTY_ITERATOR; }

 private Object readResolve() { return EMPTY_ITERATOR; }
}

/** An empty iterator. It is serializable and cloneable.
 *
 * <p>The class of this objects represent an abstract empty iterator
 * that can iterate as a type-specific (list) iterator.
 */

SUPPRESS_WARNINGS_KEY_RAWTYPES
public static final EmptyIterator EMPTY_ITERATOR = new EmptyIterator();

#if KEYS_REFERENCE
/** Returns an empty iterator. It is serializable and cloneable.
 *
 * <p>The class of the object returned represent an abstract empty iterator
 * that can iterate as a type-specific (list) iterator.
 *
 * <p>This method provides a typesafe access to {@link #EMPTY_ITERATOR}.
 * @return an empty iterator.
 */
@SuppressWarnings("unchecked")
public static KEY_GENERIC KEY_ITERATOR KEY_GENERIC emptyIterator()
 { return EMPTY_ITERATOR; }
#endif
```

```
/** An iterator returning a single element. */

private static class SingletonIterator KEY_GENERIC implements KEY_LIST_ITERATOR KEY_GENERIC {
 private final KEY_GENERIC_TYPE element;
 private int curr;

 public SingletonIterator(final KEY_GENERIC_TYPE element) {
  this.element = element;
 }

 @Override
 public boolean hasNext() { return curr == 0; }

 @Override
 public boolean hasPrevious() { return curr == 1; }

 @Override
 public KEY_GENERIC_TYPE NEXT_KEY() {
  if (! hasNext()) throw new NoSuchElementException();
  curr = 1;
  return element;
 }

 @Override
 public KEY_GENERIC_TYPE PREV_KEY() {
  if (! hasPrevious()) throw new NoSuchElementException();
  curr = 0;
  return element;
 }

 @Override
 public int nextIndex() {
  return curr;
 }

 @Override
 public int previousIndex() {
  return curr - 1;
 }
}


/** Returns an immutable iterator that iterates just over the given element.
 *
 * @param element
the only element to be returned by a type-specific list iterator.
 * @return an immutable iterator that iterates just over {@code element}.
 */
```

```
 public static KEY_GENERIC KEY_LIST_ITERATOR KEY_GENERIC singleton(final KEY_GENERIC_TYPE
element) {
  return new SingletonIterator KEY_GENERIC_DIAMOND(element);
 }


 /** A class to wrap arrays in iterators. */

 private static class ArrayIterator KEY_GENERIC implements KEY_LIST_ITERATOR KEY_GENERIC {
  private final KEY_GENERIC_TYPE[] array;
  private final int offset, length;
  private int curr;

  public ArrayIterator(final KEY_GENERIC_TYPE[] array, final int offset, final int length) {
   this.array = array;
   this.offset = offset;
   this.length = length;
  }

  @Override
  public boolean hasNext() { return curr < length; }

  @Override
  public boolean hasPrevious() { return curr > 0; }

  @Override
  public KEY_GENERIC_TYPE NEXT_KEY() {
   if (! hasNext()) throw new NoSuchElementException();
   return array[offset + curr++];
  }

  @Override
  public
 KEY_GENERIC_TYPE PREV_KEY() {
   if (! hasPrevious()) throw new NoSuchElementException();
   return array[offset + --curr];
  }

  @Override
  public int skip(int n) {
   if (n <= length - curr) {
    curr += n;
    return n;
   }
   n = length - curr;
   curr = length;
   return n;
  }
```

```
 @Override
 public int back(int n) {
  if (n <= curr) {
   curr -= n;
   return n;
  }
  n = curr;
  curr = 0;
  return n;
 }

 @Override
 public int nextIndex() {
  return curr;
 }

 @Override
 public int previousIndex() {
  return curr - 1;
 }
}


/** Wraps the given part of an array into a type-specific list iterator.
 *
 * <p>The type-specific list iterator returned by this method will iterate
 * {@code length} times, returning consecutive elements of the given
 * array starting from the one with index {@code offset}.
 *
 * @param array an array to wrap into a type-specific list iterator.
 * @param offset the first element of the array to be returned.

* @param length the number of elements to return.
 * @return an iterator that will return {@code length} elements of {@code array} starting at position {@code
offset}.
 */
 public static KEY_GENERIC KEY_LIST_ITERATOR KEY_GENERIC wrap(final KEY_GENERIC_TYPE[]
array, final int offset, final int length) {
  ARRAYS.ensureOffsetLength(array, offset, length);
  return new ArrayIterator KEY_GENERIC_DIAMOND(array, offset, length);
 }

 /** Wraps the given array into a type-specific list iterator.
  *
  * <p>The type-specific list iterator returned by this method will return
  * all elements of the given array.
  *
```

```
 * @param array an array to wrap into a type-specific list iterator.
 * @return an iterator that will the elements of {@code array}.
 */
 public static KEY_GENERIC KEY_LIST_ITERATOR KEY_GENERIC wrap(final KEY_GENERIC_TYPE[]
array) {
  return new ArrayIterator KEY_GENERIC_DIAMOND(array, 0, array.length);
 }


 /** Unwraps an iterator into an array starting at a given offset for
 a given number of elements.
 *
 * <p>This method iterates over the given type-specific iterator and stores the elements
 * returned, up to a maximum of {@code length}, in the given array starting at {@code offset}.
 * The number of actually unwrapped elements is returned (it may be less than {@code max} if
 * the iterator emits less than {@code max} elements).
 *
 * @param i a type-specific iterator.
 * @param array an array to contain the output of the iterator.
 * @param offset the first element of the array to be returned.
 * @param max the maximum number of elements to unwrap.
 * @return the number of elements unwrapped.
 */
 public static KEY_GENERIC int unwrap(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i, final
KEY_GENERIC_TYPE array[], int offset, final int max) {
  if (max < 0) throw new IllegalArgumentException("The maximum number of elements (" + max + ") is negative");
  if (offset < 0 || offset + max > array.length) throw new IllegalArgumentException();
  int
 j = max;
  while(j-- != 0 && i.hasNext()) array[offset++] = i.NEXT_KEY();
  return max - j - 1;
 }

 /** Unwraps an iterator into an array.
 *
 * <p>This method iterates over the given type-specific iterator and stores the
 * elements returned in the given array. The iteration will stop when the
 * iterator has no more elements or when the end of the array has been reached.
 *
 * @param i a type-specific iterator.
 * @param array an array to contain the output of the iterator.
 * @return the number of elements unwrapped.
 */
 public static KEY_GENERIC int unwrap(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i, final
KEY_GENERIC_TYPE array[]) {
  return unwrap(i, array, 0, array.length);
 }
```

```
/** Unwraps an iterator, returning an array, with a limit on the number of elements.
 *
 * <p>This method iterates over the given type-specific iterator and returns an array
 * containing the elements returned by the iterator. At most {@code max} elements
 * will be returned.
 *
 * @param
i a type-specific iterator.
 * @param max the maximum number of elements to be unwrapped.
 * @return an array containing the elements returned by the iterator (at most {@code max}).
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC KEY_GENERIC_TYPE[] unwrap(final STD_KEY_ITERATOR
KEY_EXTENDS_GENERIC i, int max) {
 if (max < 0) throw new IllegalArgumentException("The maximum number of elements (" + max + ") is negative");
 KEY_GENERIC_TYPE array[] = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[16];
 int j = 0;

 while(max-- != 0 && i.hasNext()) {
  if (j == array.length) array = ARRAYS.grow(array, j + 1);
  array[j++] = i.NEXT_KEY();
 }

 return ARRAYS.trim(array, j);
}


/** Unwraps an iterator, returning an array.
 *
 * <p>This method iterates over the given type-specific iterator and returns an array
 * containing the elements returned by the iterator.
 *
 * @param i a type-specific iterator.
 * @return an array containing the elements returned by the iterator.

 */

public static KEY_GENERIC KEY_GENERIC_TYPE[] unwrap(final STD_KEY_ITERATOR
KEY_EXTENDS_GENERIC i) {
 return unwrap(i, Integer.MAX_VALUE);
}


/** Unwraps an iterator into a big array starting at a given offset for a given number of elements.
 *
 * <p>This method iterates over the given type-specific iterator and stores the elements
 * returned, up to a maximum of {@code length}, in the given big array starting at {@code offset}.
 * The number of actually unwrapped elements is returned (it may be less than {@code max} if
```

```
 * the iterator emits less than {@code max} elements).
 *
 * @param i a type-specific iterator.
 * @param array a big array to contain the output of the iterator.
 * @param offset the first element of the array to be returned.
 * @param max the maximum number of elements to unwrap.
 * @return the number of elements unwrapped.
 */
 public static KEY_GENERIC long unwrap(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i, final
KEY_GENERIC_TYPE array[][],
 long offset, final long max) {
  if (max < 0) throw new IllegalArgumentException("The maximum number of elements (" + max + ") is negative");
  if (offset < 0 || offset + max > length(array)) throw new IllegalArgumentException();
  long j = max;
  while(j-- != 0 && i.hasNext()) set(array, offset++, i.NEXT_KEY());
  return max - j - 1;
 }

 /** Unwraps an iterator into a big array.
 *
 * <p>This method iterates over the given type-specific iterator and stores the
 * elements returned in the given big array. The iteration will stop when the
 * iterator has no more elements or when the end of the array has been reached.
 *
 * @param i a type-specific iterator.
 * @param array a big array to contain the output of the iterator.
 * @return the number of elements unwrapped.
 */
 public static KEY_GENERIC long unwrap(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i, final
KEY_GENERIC_TYPE array[][]) {
  return unwrap(i, array, 0, length(array));
 }

 /** Unwraps an iterator into a type-specific
 collection, with a limit on the number of elements.
 *
 * <p>This method iterates over the given type-specific iterator and stores the elements
 * returned, up to a maximum of {@code max}, in the given type-specific collection.
 * The number of actually unwrapped elements is returned (it may be less than {@code max} if
 * the iterator emits less than {@code max} elements).
 *
 * @param i a type-specific iterator.
 * @param c a type-specific collection array to contain the output of the iterator.
 * @param max the maximum number of elements to unwrap.
 * @return the number of elements unwrapped. Note that
 * this is the number of elements returned by the iterator, which is not necessarily the number
 * of elements that have been added to the collection (because of duplicates).
 */
```

```
 public static KEY_GENERIC int unwrap(final STD_KEY_ITERATOR KEY_GENERIC i, final COLLECTION
KEY_SUPER_GENERIC c, final int max) {
  if (max < 0) throw new IllegalArgumentException("The
 maximum number of elements (" + max + ") is negative");
  int j = max;
  while(j-- != 0 && i.hasNext()) c.add(i.NEXT_KEY());
  return max - j - 1;
 }

 /** Unwraps an iterator, returning a big array, with a limit on the number of elements.
  *
  * <p>This method iterates over the given type-specific iterator and returns a big array
  * containing the elements returned by the iterator. At most {@code max} elements
  * will be returned.
  *
  * @param i a type-specific iterator.
  * @param max the maximum number of elements to be unwrapped.
  * @return a big array containing the elements returned by the iterator (at most {@code max}).
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] unwrapBig(final STD_KEY_ITERATOR
KEY_EXTENDS_GENERIC i, long max) {
  if (max < 0) throw new IllegalArgumentException("The maximum number of elements (" + max + ") is negative");
  KEY_GENERIC_TYPE array[][] = KEY_GENERIC_BIG_ARRAY_CAST BIG_ARRAYS.newBigArray(16);
  long
 j = 0;

  while(max-- != 0 && i.hasNext()) {
   if (j == length(array)) array = grow(array, j + 1);
   set(array, j++, i.NEXT_KEY());
  }

  return trim(array, j);
 }


 /** Unwraps an iterator, returning a big array.
  *
  * <p>This method iterates over the given type-specific iterator and returns a big array
  * containing the elements returned by the iterator.
  *
  * @param i a type-specific iterator.
  * @return a big array containing the elements returned by the iterator.
  */

 public static KEY_GENERIC KEY_GENERIC_TYPE[][] unwrapBig(final STD_KEY_ITERATOR
KEY_EXTENDS_GENERIC i) {
  return unwrapBig(i, Long.MAX_VALUE);
```

```
 }

 /** Unwraps an iterator into a type-specific collection.
  *
  * <p>This method iterates over the given type-specific iterator and stores the
  * elements returned in the given type-specific collection. The returned count on the number
  * unwrapped elements is a long, so that it will work also with very large collections.
  *
  * @param i a type-specific iterator.

 * @param c a type-specific collection to contain the output of the iterator.
  * @return the number of elements unwrapped. Note that
  * this is the number of elements returned by the iterator, which is not necessarily the number
  * of elements that have been added to the collection (because of duplicates).
  */
 public static KEY_GENERIC long unwrap(final STD_KEY_ITERATOR KEY_GENERIC i, final COLLECTION
KEY_SUPER_GENERIC c) {
  long n = 0;
  while(i.hasNext()) {
   c.add(i.NEXT_KEY());
   n++;
  }
  return n;
 }


 /** Pours an iterator into a type-specific collection, with a limit on the number of elements.
  *
  * <p>This method iterates over the given type-specific iterator and adds
  * the returned elements to the given collection (up to {@code max}).
  *
  * @param i a type-specific iterator.
  * @param s a type-specific collection.
  * @param max the maximum number of elements to be poured.
  * @return the number of elements poured. Note that
  * this is the number of
 elements returned by the iterator, which is not necessarily the number
  * of elements that have been added to the collection (because of duplicates).
  */

 public static KEY_GENERIC int pour(final STD_KEY_ITERATOR KEY_GENERIC i, final COLLECTION
KEY_SUPER_GENERIC s, final int max) {
  if (max < 0) throw new IllegalArgumentException("The maximum number of elements (" + max + ") is negative");
  int j = max;
  while(j-- != 0 && i.hasNext()) s.add(i.NEXT_KEY());
  return max - j - 1;
 }
```

```
/** Pours an iterator into a type-specific collection.
 *
 * <p>This method iterates over the given type-specific iterator and adds
 * the returned elements to the given collection.
 *
 * @param i a type-specific iterator.
 * @param s a type-specific collection.
 * @return the number of elements poured. Note that
 * this is the number of elements returned by the iterator, which is not necessarily the number
 * of elements that have been added to the collection (because of duplicates).
 */

public
static KEY_GENERIC int pour(final STD_KEY_ITERATOR KEY_GENERIC i, final COLLECTION
KEY_SUPER_GENERIC s) {
 return pour(i, s, Integer.MAX_VALUE);
}

/** Pours an iterator, returning a type-specific list, with a limit on the number of elements.
 *
 * <p>This method iterates over the given type-specific iterator and returns
 * a type-specific list containing the returned elements (up to {@code max}). Iteration
 * on the returned list is guaranteed to produce the elements in the same order
 * in which they appeared in the iterator.
 *
 *
 * @param i a type-specific iterator.
 * @param max the maximum number of elements to be poured.
 * @return a type-specific list containing the returned elements, up to {@code max}.
 */

public static KEY_GENERIC LIST KEY_GENERIC pour(final STD_KEY_ITERATOR KEY_GENERIC i, int
max) {
 final ARRAY_LIST KEY_GENERIC l = new ARRAY_LIST KEY_GENERIC_DIAMOND();
 pour(i, l, max);
 l.trim();
 return l;
}

/** Pours an iterator, returning a
type-specific list.
 *
 * <p>This method iterates over the given type-specific iterator and returns
 * a list containing the returned elements. Iteration
 * on the returned list is guaranteed to produce the elements in the same order
 * in which they appeared in the iterator.
 *
 * @param i a type-specific iterator.
```

```
 * @return a type-specific list containing the returned elements.
 */

public static KEY_GENERIC LIST KEY_GENERIC pour(final STD_KEY_ITERATOR KEY_GENERIC i) {
 return pour(i, Integer.MAX_VALUE);
}

private static class IteratorWrapper KEY_GENERIC implements KEY_ITERATOR KEY_GENERIC {
 final Iterator<KEY_GENERIC_CLASS> i;

 public IteratorWrapper(final Iterator<KEY_GENERIC_CLASS> i) {
  this.i = i;
 }

 @Override
 public boolean hasNext() { return i.hasNext(); }
 @Override
 public void remove() { i.remove(); }
 @Override
 public KEY_GENERIC_TYPE NEXT_KEY() { return KEY_CLASS2TYPE(i.next()); }
}

/** Wraps a standard iterator into a type-specific iterator.

 *
 * <p>This method wraps a standard iterator into a type-specific one which will handle the
 * type conversions for you. Of course, any attempt to wrap an iterator returning the
 * instances of the wrong class will generate a {@link ClassCastException}. The
 * returned iterator is backed by {@code i}: changes to one of the iterators
 * will affect the other, too.
 *
 * <p>If {@code i} is already type-specific, it will returned and no new object
 * will be generated.
 *
 * @param i an iterator.
 * @return a type-specific iterator  backed by {@code i}.
 */
#if KEYS_PRIMITIVE
 @SuppressWarnings({"unchecked","rawtypes"})
#endif
 public static KEY_GENERIC KEY_ITERATOR KEY_GENERIC AS_KEY_ITERATOR(final Iterator
KEY_GENERIC i) {
  if (i instanceof KEY_ITERATOR) return (KEY_ITERATOR KEY_GENERIC)i;
  return new IteratorWrapper KEY_GENERIC_DIAMOND(i);
 }


 private static class ListIteratorWrapper KEY_GENERIC implements KEY_LIST_ITERATOR KEY_GENERIC {
```

```
 final ListIterator<KEY_GENERIC_CLASS>
i;

 public ListIteratorWrapper(final ListIterator<KEY_GENERIC_CLASS> i) {
  this.i = i;
 }

 @Override
 public boolean hasNext() { return i.hasNext(); }

 @Override
 public boolean hasPrevious() { return i.hasPrevious(); }

 @Override
 public int nextIndex() { return i.nextIndex(); }

 @Override
 public int previousIndex() { return i.previousIndex(); }

 @Override
 public void set(KEY_GENERIC_TYPE k) { i.set(KEY2OBJ(k)); }

 @Override
 public void add(KEY_GENERIC_TYPE k) { i.add(KEY2OBJ(k)); }

 @Override
 public void remove() { i.remove(); }

 @Override
 public KEY_GENERIC_TYPE NEXT_KEY() { return KEY_CLASS2TYPE(i.next()); }

 @Override
 public KEY_GENERIC_TYPE PREV_KEY() { return KEY_CLASS2TYPE(i.previous()); }
}

/** Wraps a standard list iterator into a type-specific list iterator.
 *
 * <p>This method wraps a standard list iterator into a type-specific one
 * which will handle the type conversions for you. Of course, any attempt
 * to
wrap an iterator returning the instances of the wrong class will
 * generate a {@link ClassCastException}. The
 * returned iterator is backed by {@code i}: changes to one of the iterators
 * will affect the other, too.
 *
 * <p>If {@code i} is already type-specific, it will returned and no new object
 * will be generated.
 *
```

```
  * @param i a list iterator.
  * @return a type-specific list iterator backed by {@code i}.
  */
#if KEYS_PRIMITIVE
 @SuppressWarnings({"unchecked","rawtypes"})
#endif
 public static KEY_GENERIC KEY_LIST_ITERATOR KEY_GENERIC AS_KEY_ITERATOR(final ListIterator
KEY_GENERIC i) {
  if (i instanceof KEY_LIST_ITERATOR) return (KEY_LIST_ITERATOR KEY_GENERIC)i;
  return new ListIteratorWrapper KEY_GENERIC_DIAMOND(i);
 }

#ifdef JDK_PRIMITIVE_PREDICATE
 public static boolean any(final KEY_ITERATOR iterator, final JDK_PRIMITIVE_PREDICATE predicate) {
#else
 public static KEY_GENERIC boolean any(final KEY_ITERATOR KEY_GENERIC iterator, final
java.util.function.Predicate<?
 super KEY_GENERIC_CLASS> predicate) {
#endif
  return indexOf(iterator, predicate) != -1;
 }

#ifdef JDK_PRIMITIVE_PREDICATE
 public static boolean all(final KEY_ITERATOR iterator, final JDK_PRIMITIVE_PREDICATE predicate) {
#else
 public static KEY_GENERIC boolean all(final KEY_ITERATOR KEY_GENERIC iterator, final
java.util.function.Predicate<? super KEY_GENERIC_CLASS> predicate) {
#endif
  Objects.requireNonNull(predicate);

  do {
   if (!iterator.hasNext()) return true;
#if KEY_CLASS_Boolean
  } while (predicate.test(Boolean.valueOf(iterator.nextBoolean())));
#else
  } while (predicate.test(iterator.NEXT_KEY()));
#endif
  return false;
 }

#ifdef JDK_PRIMITIVE_PREDICATE
 public static int indexOf(final KEY_ITERATOR iterator, final JDK_PRIMITIVE_PREDICATE predicate) {
#else
 public static KEY_GENERIC int indexOf(final KEY_ITERATOR KEY_GENERIC iterator, final
java.util.function.Predicate<? super KEY_GENERIC_CLASS> predicate) {
#endif
  Objects.requireNonNull(predicate);
```

```
  for
 (int i = 0; iterator.hasNext(); ++i) {
#if KEY_CLASS_Boolean
   if (predicate.test(Boolean.valueOf(iterator.nextBoolean()))) return i;
#else
   if (predicate.test(iterator.NEXT_KEY())) return i;
#endif
  }

  return -1;
 }

#if KEY_CLASS_Integer || KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character ||
KEY_CLASS_Long

#if KEY_CLASS_Long
 private static class IntervalIterator implements KEY_BIDI_ITERATOR {
#else
 private static class IntervalIterator implements KEY_LIST_ITERATOR {
#endif
  private final KEY_TYPE from, to;
  KEY_TYPE curr;

  public IntervalIterator(final KEY_TYPE from, final KEY_TYPE to) {
   this.from = this.curr = from;
   this.to = to;
  }

  @Override
  public boolean hasNext() { return curr < to; }

  @Override
  public boolean hasPrevious() { return curr > from; }

  @Override
  public KEY_TYPE NEXT_KEY() {
   if (! hasNext()) throw new NoSuchElementException();
   return curr++;
  }

  @Override
  public KEY_TYPE PREV_KEY() {
   if (! hasPrevious()) throw new
 NoSuchElementException();
   return --curr;
  }

#if ! KEY_CLASS_Long
```

```
  @Override
  public int nextIndex() { return curr - from; }

  @Override
  public int previousIndex() { return curr - from - 1; }
#endif

  @Override
  public int skip(int n) {
   if (curr + n <= to) {
    curr += n;
    return n;
   }
#if ! KEY_CLASS_Long
   n = to - curr;
#else
   n = (int)(to - curr);
#endif
   curr = to;
   return n;
  }

  @Override
  public int back(int n) {
   if (curr - n >= from) {
    curr -= n;
    return n;
   }
#if ! KEY_CLASS_Long
   n = curr - from ;
#else
   n = (int)(curr - from);
#endif
   curr = from;
   return n;
  }
 }

#if KEY_CLASS_Long
 /** Creates a type-specific bidirectional iterator over an interval.
  *
  * <p>The type-specific bidirectional iterator returned by this method will return the
  * elements {@code from}, {@code from+1},&hellip;, {@code to-1}.
  *
  * <p>Note that all other type-specific interval iterator are <em>list</em>
  * iterators. Of course, this is not
 possible with longs as the index
  * returned by {@link java.util.ListIterator#nextIndex() nextIndex()}/{@link
```

```
 * java.util.ListIterator#previousIndex() previousIndex()} would exceed an integer.
 *
 * @param from the starting element (inclusive).
 * @param to the ending element (exclusive).
 * @return a type-specific bidirectional iterator enumerating the elements from {@code from} to {@code to}.
 */
 public static KEY_BIDI_ITERATOR fromTo(final KEY_TYPE from, final KEY_TYPE to) {
  return new IntervalIterator(from, to);
 }
#else

 /** Creates a type-specific list iterator over an interval.
 *
 * <p>The type-specific list iterator returned by this method will return the
 * elements {@code from}, {@code from+1},&hellip;, {@code to-1}.
 *
 * @param from the starting element (inclusive).
 * @param to the ending element (exclusive).
 * @return a type-specific list iterator enumerating the elements from {@code from} to {@code to}.
 */
 public static KEY_LIST_ITERATOR fromTo(final
 KEY_TYPE from, final KEY_TYPE to) {
  return new IntervalIterator(from, to);
 }

#endif

#endif

 private static class IteratorConcatenator KEY_GENERIC implements KEY_ITERATOR KEY_GENERIC {
  final KEY_ITERATOR KEY_EXTENDS_GENERIC a[];
  int offset, length, lastOffset = -1;

  public IteratorConcatenator(final KEY_ITERATOR KEY_EXTENDS_GENERIC a[], int offset, int length) {
   this.a = a;
   this.offset = offset;
   this.length = length;
   advance();
  }

  private void advance() {
   while(length != 0) {
    if (a[offset].hasNext()) break;
    length--;
    offset++;
   }

   return;
```

```
 }

 @Override
 public boolean hasNext() {
  return length > 0;
 }

 @Override
 public KEY_GENERIC_TYPE NEXT_KEY() {
  if (! hasNext()) throw new NoSuchElementException();
  KEY_GENERIC_TYPE next = a[lastOffset = offset].NEXT_KEY();
  advance();
  return next;
 }

 @Override
 public void remove() {
  if (lastOffset == -1) throw new IllegalStateException();
  a[lastOffset].remove();
 }

 @Override
 public
int skip(int n) {
  lastOffset = -1;

  int skipped = 0;

  while(skipped < n && length != 0) {
   skipped += a[offset].skip(n - skipped);
   if (a[offset].hasNext()) break;
   length--;
   offset++;
  }

  return skipped;
 }
}


/** Concatenates all iterators contained in an array.
 *
 * <p>This method returns an iterator that will enumerate in order the elements returned
 * by all iterators contained in the given array.
 *
 * @param a an array of iterators.
 * @return an iterator obtained by concatenation.
 */
```

```
 public static KEY_GENERIC KEY_ITERATOR KEY_GENERIC concat(final KEY_ITERATOR
KEY_EXTENDS_GENERIC a[]) {
  return concat(a, 0, a.length);
 }


 /** Concatenates a sequence of iterators contained in an array.
  *
  * <p>This method returns an iterator that will enumerate in order the elements returned
  * by {@code a[offset]}, then those returned
  * by {@code a[offset + 1]}, and so on up to
  * {@code a[offset + length - 1]}.
  *

 * @param a an array of iterators.
  * @param offset the index of the first iterator to concatenate.
  * @param length the number of iterators to concatenate.
  * @return an iterator obtained by concatenation of {@code length} elements of {@code a} starting at {@code
offset}.
  */

 public static KEY_GENERIC KEY_ITERATOR KEY_GENERIC concat(final KEY_ITERATOR
KEY_EXTENDS_GENERIC a[], final int offset, final int length) {
  return new IteratorConcatenator KEY_GENERIC_DIAMOND(a, offset, length);
 }


  /** An unmodifiable wrapper class for iterators. */


 public static class UnmodifiableIterator KEY_GENERIC implements KEY_ITERATOR KEY_GENERIC {
  protected final KEY_ITERATOR KEY_GENERIC i;

  public UnmodifiableIterator(final KEY_ITERATOR KEY_GENERIC i) {
   this.i = i;
  }

  @Override
  public boolean hasNext() { return i.hasNext(); }

  @Override
  public KEY_GENERIC_TYPE NEXT_KEY() { return i.NEXT_KEY(); }
 }


 /** Returns an unmodifiable iterator backed by the specified iterator.
```

```
 *
 * @param i the iterator to be wrapped in an unmodifiable iterator.
 * @return an unmodifiable view of the specified iterator.
 */
public static KEY_GENERIC KEY_ITERATOR KEY_GENERIC unmodifiable(final KEY_ITERATOR
KEY_GENERIC i) { return new UnmodifiableIterator KEY_GENERIC_DIAMOND(i); }



 /** An unmodifiable wrapper class for bidirectional iterators. */

public static class UnmodifiableBidirectionalIterator KEY_GENERIC implements KEY_BIDI_ITERATOR
KEY_GENERIC {
 protected final KEY_BIDI_ITERATOR KEY_GENERIC i;

 public UnmodifiableBidirectionalIterator(final KEY_BIDI_ITERATOR KEY_GENERIC i) {
  this.i = i;
 }

 @Override
 public boolean hasNext() { return i.hasNext(); }

 @Override
 public boolean hasPrevious() { return i.hasPrevious(); }

 @Override
 public KEY_GENERIC_TYPE NEXT_KEY() { return i.NEXT_KEY(); }

 @Override
 public KEY_GENERIC_TYPE PREV_KEY() { return i.PREV_KEY(); }
}


/** Returns an unmodifiable bidirectional iterator backed by the specified
bidirectional iterator.
 *
 * @param i the bidirectional iterator to be wrapped in an unmodifiable bidirectional iterator.
 * @return an unmodifiable view of the specified bidirectional iterator.
 */
public static KEY_GENERIC KEY_BIDI_ITERATOR KEY_GENERIC unmodifiable(final
KEY_BIDI_ITERATOR KEY_GENERIC i) { return new UnmodifiableBidirectionalIterator
KEY_GENERIC_DIAMOND(i); }


 /** An unmodifiable wrapper class for list iterators. */

public static class UnmodifiableListIterator KEY_GENERIC implements KEY_LIST_ITERATOR
KEY_GENERIC {
```

```
 protected final KEY_LIST_ITERATOR KEY_GENERIC i;

 public UnmodifiableListIterator(final KEY_LIST_ITERATOR KEY_GENERIC i) {
  this.i = i;
 }

 @Override
 public boolean hasNext() { return i.hasNext(); }

 @Override
 public boolean hasPrevious() { return i.hasPrevious(); }

 @Override
 public KEY_GENERIC_TYPE NEXT_KEY() { return i.NEXT_KEY(); }

 @Override
 public KEY_GENERIC_TYPE PREV_KEY() { return i.PREV_KEY(); }

 @Override
 public
int nextIndex() { return i.nextIndex(); }

 @Override
 public int previousIndex() { return i.previousIndex(); }
}

/** Returns an unmodifiable list iterator backed by the specified list iterator.
 *
 * @param i the list iterator to be wrapped in an unmodifiable list iterator.
 * @return an unmodifiable view of the specified list iterator.
 */
 public static KEY_GENERIC KEY_LIST_ITERATOR KEY_GENERIC unmodifiable(final
KEY_LIST_ITERATOR KEY_GENERIC i) { return new UnmodifiableListIterator
KEY_GENERIC_DIAMOND(i); }

#if SMALL_TYPES && (KEY_CLASS_Short || KEY_CLASS_Integer || KEY_CLASS_Long ||
KEY_CLASS_Float || KEY_CLASS_Double)

 /** A wrapper promoting the results of a ByteIterator. */

 protected static class ByteIteratorWrapper implements KEY_ITERATOR {
  final it.unimi.dsi.fastutil.bytes.ByteIterator iterator;

  public ByteIteratorWrapper(final it.unimi.dsi.fastutil.bytes.ByteIterator iterator) {
   this.iterator = iterator;
  }

  @Override
```

```
 public boolean hasNext()
 { return iterator.hasNext(); }

 @Deprecated
 @Override
 public KEY_GENERIC_CLASS next() { return KEY_GENERIC_CLASS.valueOf(iterator.nextByte()); }

 @Override
 public KEY_TYPE NEXT_KEY() { return iterator.nextByte(); }

 @Override
 public void remove() { iterator.remove(); }

 @Override
 public int skip(final int n) { return iterator.skip(n); }
 }

 /** Returns an iterator backed by the specified byte iterator.
  * @param iterator a byte iterator.
  * @return an iterator backed by the specified byte iterator.
  */
 public static KEY_ITERATOR wrap(final it.unimi.dsi.fastutil.bytes.ByteIterator iterator) {
  return new ByteIteratorWrapper(iterator);
 }
#endif

#if SMALL_TYPES && (KEY_CLASS_Integer || KEY_CLASS_Long || KEY_CLASS_Float ||
KEY_CLASS_Double)

 /** A wrapper promoting the results of a ShortIterator. */

 protected static class ShortIteratorWrapper implements KEY_ITERATOR {
  final it.unimi.dsi.fastutil.shorts.ShortIterator iterator;

  public ShortIteratorWrapper(final
 it.unimi.dsi.fastutil.shorts.ShortIterator iterator) {
   this.iterator = iterator;
  }

 @Override
 public boolean hasNext() { return iterator.hasNext(); }

 @Deprecated
 @Override
 public KEY_GENERIC_CLASS next() { return KEY_GENERIC_CLASS.valueOf(iterator.nextShort()); }

 @Override
 public KEY_TYPE NEXT_KEY() { return iterator.nextShort(); }
```

```
 @Override
 public void remove() { iterator.remove(); }

 @Override
 public int skip(final int n) { return iterator.skip(n); }
}

/** Returns an iterator backed by the specified short iterator.
 * @param iterator a short iterator.
 * @return an iterator backed by the specified short iterator.
 */
public static KEY_ITERATOR wrap(final it.unimi.dsi.fastutil.shorts.ShortIterator iterator) {
 return new ShortIteratorWrapper(iterator);
}

#endif

#if KEY_CLASS_Long || KEY_CLASS_Double

 /** A wrapper promoting the results of an IntIterator. */

protected static class IntIteratorWrapper implements KEY_ITERATOR {
 final it.unimi.dsi.fastutil.ints.IntIterator
iterator;

 public IntIteratorWrapper(final it.unimi.dsi.fastutil.ints.IntIterator iterator) {
  this.iterator = iterator;
 }

 @Override
 public boolean hasNext() { return iterator.hasNext(); }

 @Deprecated
 @Override
 public KEY_GENERIC_CLASS next() { return KEY_GENERIC_CLASS.valueOf(iterator.nextInt()); }

 @Override
 public KEY_TYPE NEXT_KEY() { return iterator.nextInt(); }

 @Override
 public void remove() { iterator.remove(); }

 @Override
 public int skip(final int n) { return iterator.skip(n); }
}

/** Returns an iterator backed by the specified integer iterator.
```

```
 * @param iterator an integer iterator.
 * @return an iterator backed by the specified integer iterator.
 */

public static KEY_ITERATOR wrap(final it.unimi.dsi.fastutil.ints.IntIterator iterator) {
 return new IntIteratorWrapper(iterator);
}

#endif

#if SMALL_TYPES && KEY_CLASS_Double

 /** A wrapper promoting the results of a FloatIterator. */

protected
static class FloatIteratorWrapper implements KEY_ITERATOR {
 final it.unimi.dsi.fastutil.floats.FloatIterator iterator;

 public FloatIteratorWrapper(final it.unimi.dsi.fastutil.floats.FloatIterator iterator) {
  this.iterator = iterator;
 }

 @Override
 public boolean hasNext() { return iterator.hasNext(); }

 @Deprecated
 @Override
 public KEY_GENERIC_CLASS next() { return KEY_GENERIC_CLASS.valueOf(iterator.nextFloat()); }

 @Override
 public KEY_TYPE NEXT_KEY() { return iterator.nextFloat(); }

 @Override
 public void remove() { iterator.remove(); }

 @Override
 public int skip(final int n) { return iterator.skip(n); }
}

/** Returns an iterator backed by the specified float iterator.
 * @param iterator a float iterator.
 * @return an iterator backed by the specified float iterator.
 */
public static KEY_ITERATOR wrap(final it.unimi.dsi.fastutil.floats.FloatIterator iterator) {
 return new FloatIteratorWrapper(iterator);
}
#endif
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Iterators.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/IndirectPriorityQueue.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/PriorityQueue.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/AbstractPriorityQueue.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/AbstractIndirectPriorityQueue.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
```

```
*/


package PACKAGE;

import java.util.Comparator;

/** A type-specific {@link Comparator}; provides methods to compare two primitive types both as objects
* and as primitive types.
*
* <p>Note that {@code fastutil} provides a corresponding abstract class that
* can be used to implement this interface just by specifying
 the type-specific
* comparator.
*
* @see Comparator
*/

@FunctionalInterface
public interface KEY_COMPARATOR KEY_GENERIC extends Comparator<KEY_GENERIC_CLASS> {

 /** Compares its two primitive-type arguments for order. Returns a negative integer,
  * zero, or a positive integer as the first argument is less than, equal
  * to, or greater than the second.
  *
  * @see java.util.Comparator
  * @return a negative integer, zero, or a positive integer as the first
  * argument is less than, equal to, or greater than the second.
  */

 int compare(KEY_TYPE k1, KEY_TYPE k2);

 @Override
 default KEY_COMPARATOR KEY_GENERIC reversed() {
  return COMPARATORS.oppositeComparator(this);
 }

#if KEYS_PRIMITIVE
 /** {@inheritDoc}
  * <p>This implementation delegates to the corresponding type-specific method.
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default int compare(KEY_GENERIC_CLASS ok1, KEY_GENERIC_CLASS ok2) {
  return compare(ok1.KEY_VALUE(),
 ok2.KEY_VALUE());
 }
```

```
 /** Return a new comparator that first uses this comparator, then uses the second comparator
  * if this comparator compared the two elements as equal.
  *
  * @see Comparator#thenComparing(Comparator)
  */
 default KEY_COMPARATOR KEY_GENERIC thenComparing(KEY_COMPARATOR KEY_SUPER_GENERIC
second) {
  return (KEY_COMPARATOR KEY_SUPER_GENERIC & java.io.Serializable) (k1, k2) -> {
   int comp = compare(k1, k2);
   return comp == 0 ? second.compare(k1, k2) : comp;
  };
 }

 @Override
 default Comparator<KEY_GENERIC_CLASS> KEY_GENERIC thenComparing(Comparator<? super
KEY_GENERIC_CLASS> second) {
  if (second instanceof KEY_COMPARATOR) return thenComparing((KEY_COMPARATOR
KEY_SUPER_GENERIC)second);
  return Comparator.super.thenComparing(second);
 }
#endif
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Comparator.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.Comparator;
```

```
/** A class providing static methods and objects that do useful things with comparators.
*/

public final class COMPARATORS {

 private COMPARATORS() {}

 /** A type-specific comparator mimicking the natural order. */
#if KEYS_REFERENCE
 SUPPRESS_WARNINGS_KEY_UNCHECKED_RAWTYPES
 protected
 static class NaturalImplicitComparator implements Comparator, java.io.Serializable {
#else
 protected static class NaturalImplicitComparator implements KEY_COMPARATOR KEY_GENERIC,
java.io.Serializable {
#endif
  private static final long serialVersionUID = 1L;

  @Override
  public final int compare(final KEY_TYPE a, final KEY_TYPE b) {
#if KEYS_PRIMITIVE
   return KEY_CMP(a, b);
#else
   return ((Comparable)a).compareTo(b);
#endif
  }

  @Override
  public KEY_COMPARATOR reversed() { return OPPOSITE_COMPARATOR; }

  private Object readResolve() { return NATURAL_COMPARATOR; }
 };

 SUPPRESS_WARNINGS_KEY_RAWTYPES
 public static final KEY_COMPARATOR NATURAL_COMPARATOR = new NaturalImplicitComparator();

 /** A type-specific comparator mimicking the opposite of the natural order. */
#if KEYS_REFERENCE
 SUPPRESS_WARNINGS_KEY_UNCHECKED_RAWTYPES
 protected static class OppositeImplicitComparator implements Comparator, java.io.Serializable {
#else
 protected
 static class OppositeImplicitComparator implements KEY_COMPARATOR KEY_GENERIC, java.io.Serializable
{
#endif
  private static final long serialVersionUID = 1L;
```

```
  @Override
  public final int compare(final KEY_TYPE a, final KEY_TYPE b) {
#if KEYS_PRIMITIVE
   return - KEY_CMP(a, b);
#else
   return ((Comparable)b).compareTo(a);
#endif
  }
  @Override
  public KEY_COMPARATOR reversed() { return NATURAL_COMPARATOR; }

  private Object readResolve() { return OPPOSITE_COMPARATOR; }
 };

 SUPPRESS_WARNINGS_KEY_RAWTYPES
 public static final KEY_COMPARATOR OPPOSITE_COMPARATOR = new OppositeImplicitComparator();

#if KEYS_REFERENCE
 protected static class OppositeComparator KEY_GENERIC implements Comparator KEY_GENERIC,
java.io.Serializable {
#else
 protected static class OppositeComparator KEY_GENERIC implements KEY_COMPARATOR KEY_GENERIC,
java.io.Serializable {
#endif
  private static final long serialVersionUID = 1L;

  final KEY_COMPARATOR KEY_GENERIC comparator;

  protected OppositeComparator(final
 KEY_COMPARATOR KEY_GENERIC c) { comparator = c; }

  @Override
  public final int compare(final KEY_GENERIC_TYPE a, final KEY_GENERIC_TYPE b) { return
comparator.compare(b, a); }

  @Override
  public final KEY_COMPARATOR KEY_GENERIC reversed() { return comparator; }
 };

 /** Returns a comparator representing the opposite order of the given comparator.
  *
  * @param c a comparator.
  * @return a comparator representing the opposite order of {@code c}.
  */
 public static KEY_GENERIC KEY_COMPARATOR KEY_GENERIC oppositeComparator(final
KEY_COMPARATOR KEY_GENERIC c) {
  if (c instanceof OppositeComparator) return ((OppositeComparator KEY_GENERIC)c).comparator;
  return new OppositeComparator KEY_GENERIC_DIAMOND(c);
```

```
 }

#if KEYS_PRIMITIVE
 /** Returns a type-specific comparator that is equivalent to the given comparator.
  *
  * @param c a comparator, or {@code null}.
  * @return a type-specific comparator representing the order of {@code c}.
  */
 public static KEY_COMPARATOR
 AS_KEY_COMPARATOR(final Comparator<? super KEY_GENERIC_CLASS> c) {
  if (c == null || c instanceof KEY_COMPARATOR) return (KEY_COMPARATOR) c;

  return new KEY_COMPARATOR() {
    @Override
    public int compare(KEY_GENERIC_TYPE x, KEY_GENERIC_TYPE y) { return c.compare(KEY2OBJ(x),
KEY2OBJ(y)); }
    @SuppressWarnings("deprecation")
    @Override
    public int compare(KEY_GENERIC_CLASS x, KEY_GENERIC_CLASS y) { return c.compare(x, y); }
   };
 }
#endif
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Comparators.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2IntArrayMapTest.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.Iterator;
#if defined JDK_PRIMITIVE_ITERATOR && !KEY_WIDENED
import java.util.PrimitiveIterator;
#endif
#if (!defined JDK_PRIMITIVE_ITERATOR || KEY_WIDENED) && KEYS_PRIMITIVE
import java.util.Objects;
#endif
#ifdef KEYS_PRIMITIVE
import java.util.function.Consumer;
#endif

/** A type-specific
 {@link Iterator}; provides an additional method to avoid (un)boxing, and
* the possibility to skip elements.
*
* @see Iterator
*/

#if defined JDK_PRIMITIVE_ITERATOR && !KEY_WIDENED
public interface KEY_ITERATOR KEY_GENERIC extends JDK_PRIMITIVE_ITERATOR {
#else
public interface KEY_ITERATOR KEY_GENERIC extends Iterator<KEY_GENERIC_CLASS> {
#endif

#if KEYS_PRIMITIVE
 /**
  * Returns the next element as a primitive type.
```

```
  *
  * @return the next element in the iteration.
  * @see Iterator#next()
  */

#if defined JDK_PRIMITIVE_ITERATOR && !KEY_WIDENED
 @Override
#endif
 KEY_TYPE NEXT_KEY();

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default KEY_CLASS next() {
  return KEY_CLASS.valueOf(NEXT_KEY());
 }

#if !defined JDK_PRIMITIVE_ITERATOR || KEY_WIDENED
 /**
  * Performs the given action for each remaining element until all elements
  * have been processed or the action throws an exception.
  *
 @param action the action to be performed for each element.
  * @see java.util.Iterator#forEachRemaining(java.util.function.Consumer)
  * @since 8.0.0
  */
 default void forEachRemaining(final KEY_CONSUMER action) {
  Objects.requireNonNull(action);
  while (hasNext()) {
   action.accept(NEXT_KEY());
  }
 }
#endif

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default void forEachRemaining(final Consumer<? super KEY_GENERIC_CLASS> action) {
#if defined JDK_PRIMITIVE_ITERATOR && !KEY_WIDENED
  forEachRemaining((JDK_PRIMITIVE_KEY_CONSUMER) action::accept);
#else
  forEachRemaining((KEY_CONSUMER) action::accept);
#endif
 }

#endif
```

```
 /** Skips the given number of elements.
  *
  * <p>The effect of this call is exactly the same as that of calling {@link #next()} for {@code n} times (possibly
stopping if {@link #hasNext()} becomes false).
  *
  * @param n the number of elements to skip.
  * @return the number of elements
 actually skipped.
  * @see Iterator#next()
  */

 default int skip(final int n) {
  if (n < 0) throw new IllegalArgumentException("Argument must be nonnegative: " + n);
  int i = n;
  while(i-- != 0 && hasNext()) NEXT_KEY();
  return n - i - 1;
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Iterator.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.Collection;
import java.util.NoSuchElementException;
import java.util.Set;
```

```
/** A simple, brute-force implementation of a set based on a backing array.
*
* <p>The main purpose of this
* implementation is that of wrapping cleanly the brute-force approach to the storage of a very
*
 small number of items: just put them into an array and scan linearly to find an item.
*/

public class ARRAY_SET KEY_GENERIC extends ABSTRACT_SET KEY_GENERIC implements
java.io.Serializable, Cloneable {

 private static final long serialVersionUID = 1L;
 /** The backing array (valid up to {@link #size}, excluded). */
 private transient KEY_TYPE[] a;
 /** The number of valid entries in {@link #a}. */
 private int size;

 /** Creates a new array set using the given backing array. The resulting set will have as many elements as the array.
  *
  * <p>It is responsibility of the caller that the elements of {@code a} are distinct.
  *
  * @param a the backing array.
  */
 public ARRAY_SET(final KEY_TYPE[] a) {
  this.a = a;
  size = a.length;
 }

 /** Creates a new empty array set.
  */
 public ARRAY_SET() { this.a = ARRAYS.EMPTY_ARRAY; }

 /** Creates a new empty array set of given initial capacity.
  *
  * @param capacity the initial capacity.
  */
 public ARRAY_SET(final int capacity) {
 this.a = new KEY_TYPE[capacity]; }

 /** Creates a new array set copying the contents of a given collection.
  * @param c a collection.
  */
 public ARRAY_SET(COLLECTION KEY_GENERIC c) {
  this(c.size ());
  addAll(c);
 }

 /** Creates a new array set copying the contents of a given set.
```

```
 * @param c a collection.
 */
public ARRAY_SET(final Collection<? extends KEY_GENERIC_CLASS> c) {
 this(c.size());
 addAll(c);
}

/** Creates a new array set copying the contents of a given collection.
 * @param c a collection.
 */
public ARRAY_SET(SET KEY_GENERIC c) {
 this(c.size());
 int i = 0;
 for(KEY_TYPE x : c) {
  a[i] = x;
  i++;
 }
 size = i;
}

/** Creates a new array set copying the contents of a given set.
 * @param c a collection.
 */
public ARRAY_SET(final Set<? extends KEY_GENERIC_CLASS> c) {
 this(c.size());
 int i = 0;
 for(KEY_GENERIC_CLASS x: c) {
  a[i] = KEY_CLASS2TYPE(x);
  i++;
 }
 size = i;
}


/** Creates a new array set using the given backing
array and the given number of elements of the array.
 *
 * <p>It is responsibility of the caller that the first {@code size} elements of {@code a} are distinct.
 *
 * @param a the backing array.
 * @param size the number of valid elements in {@code a}.
 */
public ARRAY_SET(final KEY_TYPE[] a, final int size) {
 this.a = a;
 this.size = size;
 if (size > a.length) throw new IllegalArgumentException("The provided size (" + size + ") is larger than or equal to
the array size (" + a.length + ")");
 }
```

```
 private int findKey(final KEY_TYPE o) {
  for(int i = size; i-- != 0;) if (KEY_EQUALS(a[i], o)) return i;
  return -1;
 }

 @Override
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public KEY_ITERATOR KEY_GENERIC iterator() {
  return new KEY_ITERATOR KEY_GENERIC () {
   int next = 0;

   @Override
   public boolean hasNext() { return next < size; }

   @Override
   public KEY_GENERIC_TYPE NEXT_KEY() {
    if (! hasNext()) throw new NoSuchElementException();
    return KEY_GENERIC_CAST a[next++];
   }

   @Override
   public
 void remove() {
    final int tail = size-- - next--;
    System.arraycopy(a, next + 1, a, next, tail);
#if KEYS_REFERENCE
    a[size] = null;
#endif
   }
  };
 }

 @Override
 public boolean contains(final KEY_TYPE k) { return findKey(k) != -1; }

 @Override
 public int size() { return size; }

 @Override
 public boolean remove(final KEY_TYPE k) {
  final int pos = findKey(k);
  if (pos == -1) return false;
  final int tail = size - pos - 1;
  for(int i = 0; i < tail; i++) a[pos + i] = a[pos + i + 1];
  size--;
#if KEYS_REFERENCE
  a[size] = null;
```

```
#endif
  return true;
 }

 @Override
 public boolean add(final KEY_GENERIC_TYPE k) {
  final int pos = findKey(k);
  if (pos != -1) return false;
  if (size == a.length) {
   final KEY_TYPE[] b = new KEY_TYPE[size == 0 ? 2 : size * 2];
   for(int i = size; i-- != 0;) b[i] = a[i];
   a = b;
  }
  a[size++] = k;
  return true;
 }

 @Override
 public void clear() {
#if KEYS_REFERENCE
  java.util.Arrays.fill(a, 0,
 size, null);
#endif
  size = 0;
 }

 @Override
 public boolean isEmpty() { return size == 0; }

 /** Returns a deep copy of this set.
  *
  * <p>This method performs a deep copy of this array set; the data stored in the
  * set, however, is not cloned. Note that this makes a difference only for object keys.
  *
  *  @return a deep copy of this set.
  */
 @Override
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public ARRAY_SET KEY_GENERIC clone() {
  ARRAY_SET KEY_GENERIC c;
  try {
   c = (ARRAY_SET KEY_GENERIC)super.clone();
  }
  catch(CloneNotSupportedException cantHappen) {
   throw new InternalError();
  }
  c.a = a.clone();
  return c;
```

```
 }

 private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
  s.defaultWriteObject();
  for(int i = 0; i < size; i++) s.WRITE_KEY(a[i]);
 }

 private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
  s.defaultReadObject();
  a = new KEY_TYPE[size];
  for(int i = 0; i < size; i++) a[i]
 = s.READ_KEY();
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/ArraySet.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2005-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/io/MeasurableStream.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/io/FastBufferedOutputStream.java
*
 /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/io/MeasurableOutputStream.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/io/RepositionableStream.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/io/FastBufferedInputStream.java

* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/BinIO.drv
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/TextIO.drv
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/io/FastMultiByteArrayInputStream.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/io/InspectableFileCachedInputStream.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/io/MeasurableInputStream.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/io/FastByteArrayInputStream.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/io/FastByteArrayOutputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2010-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import it.unimi.dsi.fastutil.BigListIterator;
import it.unimi.dsi.fastutil.SafeMath;

/** A type-specific {@link BigListIterator}.
*
* @see BigListIterator
*/

public interface KEY_BIG_LIST_ITERATOR KEY_GENERIC extends KEY_BIDI_ITERATOR
KEY_GENERIC, BigListIterator<KEY_GENERIC_CLASS> {

 /**
```

```
  * Replaces
 the last element returned by {@link BigListIterator#next() next()} or
  * {@link BigListIterator#previous() previous()} with the specified element (optional operation).
  * @see java.util.ListIterator#set(Object)
  */

#if KEYS_REFERENCE
 @Override
#endif
 default void set(@SuppressWarnings("unused") final KEY_GENERIC_TYPE k) { throw new
UnsupportedOperationException(); }

 /**
  * Inserts the specified element into the list (optional operation).
  * @see java.util.ListIterator#add(Object)
  */

#if KEYS_REFERENCE
 @Override
#endif
 default void add(@SuppressWarnings("unused") final KEY_GENERIC_TYPE k)  { throw new
UnsupportedOperationException(); }

#if KEYS_PRIMITIVE

 /** Replaces the last element returned by {@link #next()} or {@link #previous()} with the specified element
(optional operation).
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default void set(final KEY_GENERIC_CLASS k) { set(k.KEY_VALUE()); }

 /** Inserts
 the specified element into the list (optional operation).
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default void add(final KEY_GENERIC_CLASS k) { add(k.KEY_VALUE()); }

#endif


 /** Skips the given number of elements.
  *
  * <p>The effect of this call is exactly the same as that of
  * calling {@link BigListIterator#next() next()} for {@code n} times (possibly stopping
  * if {@link #hasNext()} becomes false).
  *
```

```
 * @param n the number of elements to skip.
 * @return the number of elements actually skipped.
 * @see BigListIterator#next()
 */
default long skip(final long n) {
 long i = n;
 while(i-- != 0 && hasNext()) NEXT_KEY();
 return n - i - 1;
}

/** Moves back for the given number of elements.
 *
 * <p>The effect of this call is exactly the same as that of
 * calling {@link BigListIterator#previous() previous()} for {@code n} times (possibly stopping
 * if {@link #hasPrevious()} becomes false).
 *

* @param n the number of elements to skip back.
 * @return the number of elements actually skipped.
 * @see BigListIterator#previous()
 */
default long back(final long n) {
 long i = n;
 while(i-- != 0 && hasPrevious()) PREV_KEY();
 return n - i - 1;
}

/** {@inheritDoc}
 */
@Override
default int skip(int n) {
 return SafeMath.safeLongToInt(skip((long) n));
}
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/BigListIterator.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
```

```
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package PACKAGE;

import it.unimi.dsi.fastutil.objects.AbstractObjectSortedSet;
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;
import it.unimi.dsi.fastutil.objects.ObjectListIterator;
import it.unimi.dsi.fastutil.objects.ObjectSortedSet;

import VALUE_PACKAGE.VALUE_COLLECTION;
import VALUE_PACKAGE.VALUE_ABSTRACT_COLLECTION;
import
 VALUE_PACKAGE.VALUE_ITERATOR;

import java.util.Comparator;
import java.util.Iterator;
import java.util.Map;
import java.util.SortedMap;
import java.util.NoSuchElementException;

#if VALUES_PRIMITIVE
import VALUE_PACKAGE.VALUE_LIST_ITERATOR;
#endif

/** A type-specific AVL tree map with a fast, small-footprint implementation.
*
* <p>The iterators provided by the views of this class are type-specific {@linkplain
* it.unimi.dsi.fastutil.BidirectionalIterator bidirectional iterators}.
* Moreover, the iterator returned by {@code iterator()} can be safely cast
* to a type-specific {@linkplain java.util.ListIterator list iterator}.
*/

public class AVL_TREE_MAP KEY_VALUE_GENERIC extends ABSTRACT_SORTED_MAP
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable {

 /** A reference to the root entry. */
 protected transient Entry KEY_VALUE_GENERIC tree;

 /** Number of entries in this map. */
 protected int count;

 /** The first key in this map. */
```

```
protected
transient Entry KEY_VALUE_GENERIC firstEntry;

/** The last key in this map. */
protected transient Entry KEY_VALUE_GENERIC lastEntry;

/** Cached set of entries. */
protected transient ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> entries;

/** Cached set of keys. */
protected transient SORTED_SET KEY_GENERIC keys;

/** Cached collection of values. */
protected transient VALUE_COLLECTION VALUE_GENERIC values;

/** The value of this variable remembers, after a {@code put()}
 * or a {@code remove()}, whether the <em>domain</em> of the map
 * has been modified. */
protected transient boolean modified;

/** This map's comparator, as provided in the constructor. */
protected Comparator<? super KEY_GENERIC_CLASS> storedComparator;

/** This map's actual comparator; it may differ from {@link #storedComparator} because it is
 always a type-specific comparator, so it could be derived from the former by wrapping. */
protected transient KEY_COMPARATOR KEY_SUPER_GENERIC
actualComparator;

private static final long serialVersionUID = -7046029254386353129L;

{
 allocatePaths();
}

/** Creates a new empty tree map.
 */

public AVL_TREE_MAP() {
 tree = null;
 count = 0;
}

/** Generates the comparator that will be actually used.
 *
 * <p>When a given {@link Comparator} is specified and stored in {@link
 * #storedComparator}, we must check whether it is type-specific.  If it is
 * so, we can used directly, and we store it in {@link #actualComparator}. Otherwise,
 * we adapt it using a helper static method.
```

```
  */
 private void setActualComparator() {
#if KEY_CLASS_Object
  actualComparator = storedComparator;
#else
  actualComparator = COMPARATORS.AS_KEY_COMPARATOR(storedComparator);
#endif
 }


 /** Creates a new empty tree map with the given comparator.
  *
  * @param c a (possibly type-specific) comparator.
  */

 public AVL_TREE_MAP(final Comparator<? super KEY_GENERIC_CLASS> c) {
  this();
  storedComparator = c;
  setActualComparator();
 }


 /**
 Creates a new tree map copying a given map.
  *
  * @param m a {@link Map} to be copied into the new tree map.
  */

 public AVL_TREE_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS>
m) {
  this();
  putAll(m);
 }

 /** Creates a new tree map copying a given sorted map (and its {@link Comparator}).
  *
  * @param m a {@link SortedMap} to be copied into the new tree map.
  */

 public AVL_TREE_MAP(final SortedMap<KEY_GENERIC_CLASS,VALUE_GENERIC_CLASS> m) {
  this(m.comparator());
  putAll(m);
 }

 /** Creates a new tree map copying a given map.
  *
  * @param m a type-specific map to be copied into the new tree map.
  */
```

```
public AVL_TREE_MAP(final MAP KEY_VALUE_EXTENDS_GENERIC m) {
 this();
 putAll(m);
}

/** Creates a new tree map copying a given sorted map (and its {@link Comparator}).
 *
 * @param m a type-specific sorted map to be copied into the new tree map.
 */

public AVL_TREE_MAP(final SORTED_MAP KEY_VALUE_GENERIC m) {
 this(m.comparator());
 putAll(m);
}

/**
Creates a new tree map using the elements of two parallel arrays and the given comparator.
 *
 * @param k the array of keys of the new tree map.
 * @param v the array of corresponding values in the new tree map.
 * @param c a (possibly type-specific) comparator.
 * @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
 */

public AVL_TREE_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE v[], final
Comparator<? super KEY_GENERIC_CLASS> c) {
 this(c);
 if (k.length != v.length) throw new IllegalArgumentException("The key array and the value array have different
lengths (" + k.length + " and " + v.length + ")");
 for(int i = 0; i < k.length; i++) this.put(k[i], v[i]);
}

/** Creates a new tree map using the elements of two parallel arrays.
 *
 * @param k the array of keys of the new tree map.
 * @param v the array of corresponding values in the new tree map.
 * @throws IllegalArgumentException if {@code
k} and {@code v} have different lengths.
 */

public AVL_TREE_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE v[]) {
 this(k, v, null);
}

/*
 * The following methods implements some basic building blocks used by
 * all accessors.  They are (and should be maintained) identical to those used in AVLTreeSet.drv.
```

```
 *
 * The put()/remove() code is derived from Ben Pfaff's GNU libavl
 * (http://www.msu.edu/~pfaffben/avl/). If you want to understand what's
 * going on, you should have a look at the literate code contained therein
 * first.
 */


/** Compares two keys in the right way.
 *
 * <p>This method uses the {@link #actualComparator} if it is non-{@code null}.
 * Otherwise, it resorts to primitive type comparisons or to {@link Comparable#compareTo(Object) compareTo()}.
 *
 * @param k1 the first key.
 * @param k2 the second key.
 * @return a number smaller than, equal to or greater than 0, as usual
 * (i.e., when k1 &lt; k2, k1 = k2 or k1 &gt; k2, respectively).

*/

SUPPRESS_WARNINGS_KEY_UNCHECKED
final int compare(final KEY_GENERIC_TYPE k1, final KEY_GENERIC_TYPE k2) {
 return actualComparator == null ? KEY_CMP(k1, k2) : actualComparator.compare(k1, k2);
}



/** Returns the entry corresponding to the given key, if it is in the tree; {@code null}, otherwise.
 *
 * @param k the key to search for.
 * @return the corresponding entry, or {@code null} if no entry with the given key exists.
 */

final Entry KEY_VALUE_GENERIC findKey(final KEY_GENERIC_TYPE k) {
 Entry KEY_VALUE_GENERIC e = tree;
 int cmp;

 while (e != null && (cmp = compare(k, e.key)) != 0) e = cmp < 0 ? e.left() : e.right();

 return e;
}

/** Locates a key.
 *
 * @param k a key.
 * @return the last entry on a search for the given key; this will be
 * the given key, if it present; otherwise, it will be either the smallest greater key or the greatest smaller key.
 */
```

```
final Entry KEY_VALUE_GENERIC locateKey(final KEY_GENERIC_TYPE k) {
 Entry KEY_VALUE_GENERIC
e = tree, last = tree;
 int cmp = 0;

 while (e != null && (cmp = compare(k, e.key)) != 0) {
  last = e;
  e = cmp < 0 ? e.left() : e.right();
 }

 return cmp == 0 ? e : last;
}

/** This vector remembers the directions followed during
 * the current insertion. It suffices for about 2<sup>32</sup> entries. */
private transient boolean dirPath[];

private void allocatePaths() {
 dirPath = new boolean[48];
}

#if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
/** Adds an increment to value currently associated with a key.
*
* <p>Note that this method respects the {@linkplain #defaultReturnValue() default return value} semantics: when
* called with a key that does not currently appears in the map, the key
* will be associated with the default return value plus
* the given increment.
*
* @param k the key.
* @param incr the increment.
 * @return the old value, or the {@linkplain #defaultReturnValue() default return value} if no value was present for
the given key.
 */
 public
 VALUE_GENERIC_TYPE addTo(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE incr) {
  Entry KEY_VALUE_GENERIC e = add(k);
  final VALUE_GENERIC_TYPE oldValue = e.value;
  e.value += incr;
  return oldValue;
 }
#endif

@Override
public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
 Entry KEY_VALUE_GENERIC e = add(k);
 final VALUE_GENERIC_TYPE oldValue = e.value;
```

```
 e.value = v;
 return oldValue;
}

/** Returns a node with key k in the balanced tree, creating one with defRetValue if necessary.
*
* @param k the key
* @return a node with key k. If a node with key k already exists, then that node is returned,
*     otherwise a new node with defRetValue is created ensuring that the tree is balanced
    after creation of the node.
*/
private Entry KEY_VALUE_GENERIC add(final KEY_GENERIC_TYPE k) {
 /* After execution of this method, modified is true iff a new entry has
 been inserted. */
 modified = false;

 Entry KEY_VALUE_GENERIC
e = null;
 if (tree == null) { // The case of the empty tree is treated separately.
  count++;
  e = tree = lastEntry = firstEntry = new Entry KEY_VALUE_GENERIC_DIAMOND(k, defRetValue);
  modified = true;
 }
 else {
  Entry KEY_VALUE_GENERIC p = tree, q = null, y = tree, z = null, w = null;
  int cmp, i = 0;

  while(true) {
   if ((cmp = compare(k, p.key)) == 0) {
    return p;
   }

   if (p.balance() != 0) {
    i = 0;
    z = q;
    y = p;
   }

   if (dirPath[i++] = cmp > 0) {
    if (p.succ()) {
     count++;
     e = new Entry KEY_VALUE_GENERIC_DIAMOND(k, defRetValue);

     modified = true;
     if (p.right == null) lastEntry = e;

     e.left = p;
     e.right = p.right;
```

```
    p.right(e);

    break;
   }

   q = p;
   p = p.right;
  }
  else {
   if (p.pred()) {
    count++;
    e = new Entry KEY_VALUE_GENERIC_DIAMOND(k, defRetValue);

    modified = true;
    if (p.left == null) firstEntry = e;

    e.right = p;
    e.left
= p.left;

    p.left(e);

    break;
   }

   q = p;
   p = p.left;
  }
 }

 p = y;
 i = 0;

 while(p != e) {
  if (dirPath[i]) p.incBalance();
  else p.decBalance();

  p = dirPath[i++] ? p.right : p.left;
 }

 if (y.balance() == -2) {
  Entry KEY_VALUE_GENERIC x = y.left;

  if (x.balance() == -1) {
   w = x;
   if (x.succ()) {
    x.succ(false);
```

```
     y.pred(x);
    }
    else y.left = x.right;

    x.right = y;
    x.balance(0);
    y.balance(0);
   }
   else {
    assert x.balance() == 1;

    w = x.right;
    x.right = w.left;
    w.left = x;
    y.left = w.right;
    w.right = y;
    if (w.balance() == -1) {
     x.balance(0);
     y.balance(1);
    }
    else if (w.balance() == 0) {
     x.balance(0);
     y.balance(0);
    }
    else {
     x.balance(-1);
     y.balance(0);
    }
    w.balance(0);


    if (w.pred()) {
     x.succ(w);
     w.pred(false);
    }
    if (w.succ()) {
     y.pred(w);
     w.succ(false);
    }

   }
  }
  else
if (y.balance() == +2) {
   Entry KEY_VALUE_GENERIC x = y.right;

   if (x.balance() == 1) {
    w = x;
```

```
 if (x.pred()) {
  x.pred(false);
  y.succ(x);
 }
 else y.right = x.left;

 x.left = y;
 x.balance(0);
 y.balance(0);
}
else {
 assert x.balance() == -1;

 w = x.left;
 x.left = w.right;
 w.right = x;
 y.right = w.left;
 w.left = y;
 if (w.balance() == 1) {
  x.balance(0);
  y.balance(-1);
 }
 else if (w.balance() == 0) {
  x.balance(0);
  y.balance(0);
 }
 else {
  x.balance(1);
  y.balance(0);
 }
 w.balance(0);


 if (w.pred()) {
  y.succ(w);
  w.pred(false);
 }
 if (w.succ()) {
  x.pred(w);
  w.succ(false);
 }

 }
}
else return e;

if (z == null) tree = w;
else {
```

```
   if (z.left == y) z.left = w;
   else z.right = w;
  }
 }

 return e;
}

/** Finds the parent of an entry.

*
 * @param e a node of the tree.
 * @return the parent of the given node, or {@code null} for the root.
 */

private Entry KEY_VALUE_GENERIC parent(final Entry KEY_VALUE_GENERIC e) {
 if (e == tree) return null;

 Entry KEY_VALUE_GENERIC x, y, p;
 x = y = e;

 while(true) {
  if (y.succ()) {
   p = y.right;
   if (p == null || p.left != e) {
    while(! x.pred()) x = x.left;
    p = x.left;
   }
   return p;
  }
  else if (x.pred()) {
   p = x.left;
   if (p == null || p.right != e) {
    while(! y.succ()) y = y.right;
    p = y.right;
   }
   return p;
  }

  x = x.left;
  y = y.right;
 }
}


/* After execution of this method, {@link #modified} is true iff an entry
has been deleted. */
```

```
SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) {
 modified = false;

 if (tree == null) return defRetValue;

 int cmp;
 Entry KEY_VALUE_GENERIC p = tree, q = null;
 boolean dir = false;
 final KEY_GENERIC_TYPE
kk = KEY_GENERIC_CAST k;

 while(true) {
  if ((cmp = compare(kk, p.key)) == 0) break;
  else if (dir = cmp > 0) {
   q = p;
   if ((p = p.right()) == null) return defRetValue;
  }
  else {
   q = p;
   if ((p = p.left()) == null) return defRetValue;
  }
 }

 if (p.left == null) firstEntry = p.next();
 if (p.right == null) lastEntry = p.prev();

 if (p.succ()) {
  if (p.pred()) {
   if (q != null) {
    if (dir) q.succ(p.right);
    else q.pred(p.left);
   }
   else tree = dir ? p.right : p.left;
  }
  else {
   p.prev().right = p.right;

   if (q != null) {
    if (dir) q.right = p.left;
    else q.left = p.left;
   }
   else tree = p.left;
  }
 }
 else {
  Entry KEY_VALUE_GENERIC r = p.right;
```

```
 if (r.pred()) {
  r.left = p.left;
  r.pred(p.pred());
  if (! r.pred()) r.prev().right = r;
  if (q != null) {
   if (dir) q.right = r;
   else q.left = r;
  }
  else tree = r;

  r.balance(p.balance());
  q = r;
  dir = true;

 }
 else
{
  Entry KEY_VALUE_GENERIC s;

  while(true) {
   s = r.left;
   if (s.pred()) break;
   r = s;
  }

  if (s.succ()) r.pred(s);
  else r.left = s.right;

  s.left = p.left;

  if (! p.pred()) {
   p.prev().right = s;
   s.pred(false);
  }

  s.right = p.right;
  s.succ(false);

  if (q != null) {
   if (dir) q.right = s;
   else q.left = s;
  }
  else tree = s;

  s.balance(p.balance());
  q = r;
  dir = false;
```

```
 }
}

Entry KEY_VALUE_GENERIC y;

while(q != null) {
 y = q;
 q = parent(y);

 if (! dir) {
  dir = q != null && q.left != y;
  y.incBalance();

  if (y.balance() == 1) break;
  else if (y.balance() == 2) {

   Entry KEY_VALUE_GENERIC x = y.right;
   assert x != null;

   if (x.balance() == -1) {
    Entry KEY_VALUE_GENERIC w;

    assert x.balance() == -1;

    w = x.left;
    x.left = w.right;
    w.right = x;
    y.right = w.left;
    w.left = y;

    if (w.balance() == 1) {
     x.balance(0);
     y.balance(-1);
    }
    else
if (w.balance() == 0) {
     x.balance(0);
     y.balance(0);
    }
    else {
     assert w.balance() == -1;

     x.balance(1);
     y.balance(0);
    }

    w.balance(0);
```

```
    if (w.pred()) {
     y.succ(w);
     w.pred(false);
    }
    if (w.succ()) {
     x.pred(w);
     w.succ(false);
    }

    if (q != null) {
     if (dir) q.right = w;
     else q.left = w;
    }
    else tree = w;
   }
   else {
    if (q != null) {
     if (dir) q.right = x;
     else q.left = x;
    }
    else tree = x;

    if (x.balance() == 0) {
     y.right = x.left;
     x.left = y;
     x.balance(-1);
     y.balance(+1);
     break;
    }
    assert x.balance() == 1;

    if (x.pred()) {
     y.succ(true);
     x.pred(false);
    }
    else y.right = x.left;

    x.left = y;
    y.balance(0);
    x.balance(0);
   }
  }
 }
 else {
  dir = q != null
&& q.left != y;
  y.decBalance();
```

```
if (y.balance() == -1) break;
else if (y.balance() == -2) {

 Entry KEY_VALUE_GENERIC x = y.left;
 assert x != null;

 if (x.balance() == 1) {
  Entry KEY_VALUE_GENERIC w;

  assert x.balance() == 1;

  w = x.right;
  x.right = w.left;
  w.left = x;
  y.left = w.right;
  w.right = y;

  if (w.balance() == -1) {
   x.balance(0);
   y.balance(1);
  }
  else if (w.balance() == 0) {
   x.balance(0);
   y.balance(0);
  }
  else {
   assert w.balance() == 1;

   x.balance(-1);
   y.balance(0);
  }

  w.balance(0);

  if (w.pred()) {
   x.succ(w);
   w.pred(false);
  }
  if (w.succ()) {
   y.pred(w);
   w.succ(false);
  }

  if (q != null) {
   if (dir) q.right = w;
   else q.left = w;
  }
  else tree = w;
```

```
    }
    else {
     if (q != null) {
      if (dir) q.right = x;
      else q.left = x;
     }
     else
tree = x;

     if (x.balance() == 0) {
      y.left = x.right;
      x.right = y;
      x.balance(+1);
      y.balance(-1);
      break;
     }
     assert x.balance() == -1;

     if (x.succ()) {
      y.pred(true);
      x.succ(false);
     }
     else y.left = x.right;

     x.right = y;
     y.balance(0);
     x.balance(0);
    }
   }
  }
 }

 modified = true;
 count--;
 return p.value;
}


@Override
public boolean containsValue(final VALUE_TYPE v) {
 final ValueIterator i = new ValueIterator();
 VALUE_GENERIC_TYPE ev;

 int j = count;
 while(j-- != 0) {
  ev = i.NEXT_VALUE();
  if (VALUE_EQUALS(ev, v)) return true;
 }
```

```
 return false;
}

@Override
public void clear() {
 count = 0;
 tree = null;
 entries = null;
 values = null;
 keys = null;
 firstEntry = lastEntry = null;
}


/** This class represent an entry in a tree map.
 *
 * <p>We use the only "metadata", i.e., {@link Entry#info}, to store
 * information about
balance, predecessor status and successor status.
 *
 * <p>Note that since the class is recursive, it can be
 * considered equivalently a tree.
 */

private static final class Entry KEY_VALUE_GENERIC extends ABSTRACT_MAP.BasicEntry
KEY_VALUE_GENERIC implements Cloneable {
 /** If the bit in this mask is true, {@link #right} points to a successor. */
 private static final int SUCC_MASK = 1 << 31;
 /** If the bit in this mask is true, {@link #left} points to a predecessor. */
 private static final int PRED_MASK = 1 << 30;
 /** The bits in this mask hold the node balance info. You can get it just by casting to byte. */
 private static final int BALANCE_MASK = 0xFF;
 /** The pointers to the left and right subtrees. */
 Entry KEY_VALUE_GENERIC left, right;
 /** This integers holds different information in different bits (see {@link #SUCC_MASK}, {@link
#PRED_MASK} and {@link #BALANCE_MASK}). */
 int info;

 Entry() {
  super(KEY_NULL, VALUE_NULL);
 }

 /** Creates a new
entry with the given key and value.
  *
  * @param k a key.
  * @param v a value.
```

```
 */
Entry(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
 super(k, v);
 info = SUCC_MASK | PRED_MASK;
}

/** Returns the left subtree.
 *
 * @return the left subtree ({@code null} if the left
 * subtree is empty).
 */
Entry KEY_VALUE_GENERIC left() {
 return (info & PRED_MASK) != 0 ? null : left;
}

/** Returns the right subtree.
 *
 * @return the right subtree ({@code null} if the right
 * subtree is empty).
 */
Entry KEY_VALUE_GENERIC right() {
 return (info & SUCC_MASK) != 0 ? null : right;
}

/** Checks whether the left pointer is really a predecessor.
 * @return true if the left pointer is a predecessor.
 */
boolean pred() {
 return (info & PRED_MASK) != 0;
}

/** Checks whether the right pointer is really a successor.
 * @return true if the right pointer is a successor.
 */
boolean succ() {
 return (info & SUCC_MASK) != 0;
}

/**
Sets whether the left pointer is really a predecessor.
 * @param pred if true then the left pointer will be considered a predecessor.
 */
void pred(final boolean pred) {
 if (pred) info |= PRED_MASK;
 else info &= ~PRED_MASK;
}

/** Sets whether the right pointer is really a successor.
```

```
 * @param succ if true then the right pointer will be considered a successor.
 */
void succ(final boolean succ) {
 if (succ) info |= SUCC_MASK;
 else info &= ~SUCC_MASK;
}

/** Sets the left pointer to a predecessor.
 * @param pred the predecessr.
 */
void pred(final Entry KEY_VALUE_GENERIC pred) {
 info |= PRED_MASK;
 left = pred;
}

/** Sets the right pointer to a successor.
 * @param succ the successor.
 */
void succ(final Entry KEY_VALUE_GENERIC succ) {
 info |= SUCC_MASK;
 right = succ;
}

/** Sets the left pointer to the given subtree.
 * @param left the new left subtree.
 */
void left(final Entry KEY_VALUE_GENERIC left) {
 info
&= ~PRED_MASK;
 this.left = left;
}

/** Sets the right pointer to the given subtree.
 * @param right the new right subtree.
 */
void right(final Entry KEY_VALUE_GENERIC right) {
 info &= ~SUCC_MASK;
 this.right = right;
}

/** Returns the current level of the node.
 * @return the current level of this node.
 */
int balance() {
 return (byte)info;
}

/** Sets the level of this node.
```

```
 * @param level the new level of this node.
 */
void balance(int level) {
 info &= ~BALANCE_MASK;
 info |= (level & BALANCE_MASK);
}

/** Increments the level of this node. */
void incBalance() {
 info = info & ~BALANCE_MASK | ((byte)info + 1) & 0xFF;
}

/** Decrements the level of this node. */
protected void decBalance() {
 info = info & ~BALANCE_MASK | ((byte)info - 1) & 0xFF;
}

/** Computes the next entry in the set order.
 *
 * @return the next entry ({@code null}) if this is the last entry).
 */

Entry KEY_VALUE_GENERIC next()
{
 Entry KEY_VALUE_GENERIC next = this.right;
 if ((info & SUCC_MASK) == 0) while ((next.info & PRED_MASK) == 0) next = next.left;
 return next;
}

/** Computes the previous entry in the set order.
 *
 * @return the previous entry ({@code null}) if this is the first entry).
 */

Entry KEY_VALUE_GENERIC prev() {
 Entry KEY_VALUE_GENERIC prev = this.left;
 if ((info & PRED_MASK) == 0) while ((prev.info & SUCC_MASK) == 0) prev = prev.right;
 return prev;
}

@Override
public VALUE_GENERIC_TYPE setValue(final VALUE_GENERIC_TYPE value) {
 final VALUE_GENERIC_TYPE oldValue = this.value;
 this.value = value;
 return oldValue;
}

@Override
```

```
 SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
 public Entry KEY_VALUE_GENERIC clone() {
  Entry KEY_VALUE_GENERIC c;
  try {
   c = (Entry KEY_VALUE_GENERIC)super.clone();
  }
  catch(CloneNotSupportedException cantHappen) {
   throw new InternalError();
  }

  c.key = key;
  c.value = value;
  c.info = info;

  return
 c;
  }

  @Override
  @SuppressWarnings("unchecked")
  public boolean equals(final Object o) {
  if (!(o instanceof Map.Entry)) return false;
  Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS> e = (Map.Entry<KEY_GENERIC_CLASS,
VALUE_GENERIC_CLASS>)o;

  return KEY_EQUALS(key, KEY_CLASS2TYPE(e.getKey())) && VALUE_EQUALS(value,
VALUE_CLASS2TYPE(e.getValue()));
  }

  @Override
  public int hashCode() {
  return KEY2JAVAHASH_NOT_NULL(key) ^ VALUE2JAVAHASH(value);
  }


  @Override
  public String toString() {
  return key + "=>" + value;
  }


  /*
  public void prettyPrint() {
  prettyPrint(0);
  }

  public void prettyPrint(int level) {
  if (pred()) {
```

```
   for (int i = 0; i < level; i++)
    System.err.print("  ");
   System.err.println("pred: " + left);
  }
  else if (left != null)
   left.prettyPrint(level +1);
  for (int i = 0; i < level; i++)
   System.err.print("  ");
  System.err.println(key + "=" + value + " (" + balance() + ")");
  if (succ()) {
   for (int
i = 0; i < level; i++)
    System.err.print("  ");
   System.err.println("succ: " + right);
  }
  else if (right != null)
   right.prettyPrint(level + 1);
 }
 */
}

/*
public void prettyPrint() {
 System.err.println("size: " + count);
 if (tree != null) tree.prettyPrint();
}
*/

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public boolean containsKey(final KEY_TYPE k) {
 return findKey(KEY_GENERIC_CAST k) != null;
}

@Override
public int size() {
 return count;
}

@Override
public boolean isEmpty() {
 return count == 0;
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
 final Entry KEY_VALUE_GENERIC e = findKey(KEY_GENERIC_CAST k);
```

```
 return e == null ? defRetValue : e.value;
}

@Override
public KEY_GENERIC_TYPE FIRST_KEY() {
 if (tree == null) throw new NoSuchElementException();
 return firstEntry.key;
}

@Override
public KEY_GENERIC_TYPE LAST_KEY() {
 if (tree == null) throw new NoSuchElementException();
 return
lastEntry.key;
}


/** An abstract iterator on the whole range.
 *
 * <p>This class can iterate in both directions on a threaded tree.
 */

private class TreeIterator {
 /** The entry that will be returned by the next call to {@link java.util.ListIterator#previous()} (or {@code null} if
no previous entry exists). */
 Entry KEY_VALUE_GENERIC prev;
 /** The entry that will be returned by the next call to {@link java.util.ListIterator#next()} (or {@code null} if no
next entry exists). */
 Entry KEY_VALUE_GENERIC next;
 /** The last entry that was returned (or {@code null} if we did not iterate or used {@link #remove()}). */
 Entry KEY_VALUE_GENERIC curr;
 /** The current index (in the sense of a {@link java.util.ListIterator}). Note that this value is not meaningful when
this {@link TreeIterator} has been created using the nonempty constructor.*/
 int index = 0;

 TreeIterator() {
  next = firstEntry;
 }

 TreeIterator(final KEY_GENERIC_TYPE k) {
  if
((next = locateKey(k)) != null) {
   if (compare(next.key, k) <= 0) {
    prev = next;
    next = next.next();
   }
   else prev = next.prev();
  }
```

```
}

public boolean hasNext() { return next != null; }

public boolean hasPrevious() { return prev != null; }

void updateNext() {
 next = next.next();
}

Entry KEY_VALUE_GENERIC nextEntry() {
 if (! hasNext()) throw new NoSuchElementException();
 curr = prev = next;
 index++;
 updateNext();
 return curr;
}

void updatePrevious() {
 prev = prev.prev();
}

Entry KEY_VALUE_GENERIC previousEntry() {
 if (! hasPrevious()) throw new NoSuchElementException();
 curr = next = prev;
 index--;
 updatePrevious();
 return curr;
}

public int nextIndex() {
 return index;
}

public int previousIndex() {
 return index - 1;
}

public void remove() {
 if (curr == null) throw new IllegalStateException();
 /* If the last operation was a next(), we are removing an entry that preceeds
    the
current index, and thus we must decrement it. */
 if (curr == prev) index--;
 next = prev = curr;
 updatePrevious();
 updateNext();
 AVL_TREE_MAP.this.REMOVE_VALUE(curr.key);
```

```
  curr = null;
 }

 public int skip(final int n) {
  int i = n;
  while(i-- != 0 && hasNext()) nextEntry();
  return n - i - 1;
 }

 public int back(final int n) {
  int i = n;
  while(i-- != 0 && hasPrevious()) previousEntry();
  return n - i - 1;
 }
}


/** An iterator on the whole range.
 *
 * <p>This class can iterate in both directions on a threaded tree.
 */

private class EntryIterator extends TreeIterator implements ObjectListIterator<MAP.Entry
KEY_VALUE_GENERIC> {
 EntryIterator() {}

 EntryIterator(final KEY_GENERIC_TYPE k) {
  super(k);
 }

 @Override
 public MAP.Entry KEY_VALUE_GENERIC next() { return nextEntry(); }

 @Override
 public MAP.Entry KEY_VALUE_GENERIC previous() { return previousEntry(); }

 @Override
 public void set(MAP.Entry KEY_VALUE_GENERIC ok) {
throw new UnsupportedOperationException(); }

 @Override
 public void add(MAP.Entry KEY_VALUE_GENERIC ok) { throw new UnsupportedOperationException(); }
}


@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() {
 if (entries == null) entries = new AbstractObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC>() {
```

```
    final Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator = (Comparator<MAP.Entry
KEY_VALUE_GENERIC>) (x, y) -> AVL_TREE_MAP.this.actualComparator.compare(x.ENTRY_GET_KEY(),
y.ENTRY_GET_KEY());

    @Override
    public Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator() { return comparator; }

    @Override
    public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() { return new EntryIterator(); }

    @Override
    public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator(final MAP.Entry
KEY_VALUE_GENERIC from) { return new EntryIterator(from.ENTRY_GET_KEY()); }

    @Override
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public
 boolean contains(final Object o) {
     if (!(o instanceof Map.Entry)) return false;
     final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
     if (e.getKey() == null || ! (e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
     if (e.getValue() == null || ! (e.getValue() instanceof VALUE_CLASS)) return false;
#endif
     final Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey()));
     return e.equals(f);
    }

    @Override
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public boolean remove(final Object o) {
     if (!(o instanceof Map.Entry)) return false;
     final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
     if (e.getKey() == null || ! (e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
     if (e.getValue() == null || ! (e.getValue() instanceof VALUE_CLASS)) return false;
#endif
     final Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey()));
     if
 (f == null || ! VALUE_EQUALS(f.ENTRY_GET_VALUE(), VALUE_OBJ2TYPE(e.getValue()))) return false;
     AVL_TREE_MAP.this.REMOVE_VALUE(f.key);
     return true;
    }
```

```
    @Override
    public int size() { return count; }

    @Override
    public void clear() { AVL_TREE_MAP.this.clear(); }

    @Override
    public MAP.Entry KEY_VALUE_GENERIC first() { return firstEntry; }

    @Override
    public MAP.Entry KEY_VALUE_GENERIC last() { return lastEntry; }

    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> subSet(MAP.Entry KEY_VALUE_GENERIC
from, MAP.Entry KEY_VALUE_GENERIC to) { return subMap(from.ENTRY_GET_KEY(),
to.ENTRY_GET_KEY()).ENTRYSET(); }

    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> headSet(MAP.Entry KEY_VALUE_GENERIC
to) { return headMap(to.ENTRY_GET_KEY()).ENTRYSET(); }

    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> tailSet(MAP.Entry KEY_VALUE_GENERIC
from) { return tailMap(from.ENTRY_GET_KEY()).ENTRYSET();
  }
    };

  return entries;
  }

  /** An iterator on the whole range of keys.
   *
   * <p>This class can iterate in both directions on the keys of a threaded tree. We
   * simply override the {@link java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()} methods (and
possibly
   * their type-specific counterparts) so that they return keys instead of entries.
   */
  private final class KeyIterator extends TreeIterator implements KEY_LIST_ITERATOR KEY_GENERIC {
   public KeyIterator() {}
   public KeyIterator(final KEY_GENERIC_TYPE k) { super(k); }
   @Override
   public KEY_GENERIC_TYPE NEXT_KEY() { return nextEntry().key; }
   @Override
   public KEY_GENERIC_TYPE PREV_KEY() { return previousEntry().key; }
  }

  /** A keyset implementation using a more direct implementation for iterators. */
  private class KeySet extends ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC.KeySet {
```

```
 @Override
 public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new KeyIterator(); }
 @Override
 public
 KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
KeyIterator(from); }
 }

 /** Returns a type-specific sorted set view of the keys contained in this map.
  *
  * <p>In addition to the semantics of {@link java.util.Map#keySet()}, you can
  * safely cast the set returned by this call to a type-specific sorted
  * set interface.
  *
  * @return a type-specific sorted set view of the keys contained in this map.
  */
 @Override
 public SORTED_SET KEY_GENERIC keySet() {
  if (keys == null) keys = new KeySet();
  return keys;
 }

 /** An iterator on the whole range of values.
  *
  * <p>This class can iterate in both directions on the values of a threaded tree. We
  * simply override the {@link java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()} methods (and
possibly
  * their type-specific counterparts) so that they return values instead of entries.
  */
 private final class ValueIterator extends TreeIterator implements VALUE_LIST_ITERATOR
 VALUE_GENERIC {
  @Override
  public VALUE_GENERIC_TYPE NEXT_VALUE() { return nextEntry().value; }
  @Override
  public VALUE_GENERIC_TYPE PREV_VALUE() { return previousEntry().value; }
 }

 /** Returns a type-specific collection view of the values contained in this map.
  *
  * <p>In addition to the semantics of {@link java.util.Map#values()}, you can
  * safely cast the collection returned by this call to a type-specific collection
  * interface.
  *
  * @return a type-specific collection view of the values contained in this map.
  */
 @Override
 public VALUE_COLLECTION VALUE_GENERIC values() {
  if (values == null) values = new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {
```

```
    @Override
    public VALUE_ITERATOR VALUE_GENERIC iterator() { return new ValueIterator(); }
    @Override
    public boolean contains(final VALUE_TYPE k) { return containsValue(k); }
    @Override
    public int size() { return count; }
    @Override
    public void clear() { AVL_TREE_MAP.this.clear();
  }
   };

  return values;
  }

  @Override
  public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return actualComparator; }

  @Override
  public SORTED_MAP KEY_VALUE_GENERIC headMap(KEY_GENERIC_TYPE to) { return new
Submap(KEY_NULL, true, to, false); }

  @Override
  public SORTED_MAP KEY_VALUE_GENERIC tailMap(KEY_GENERIC_TYPE from) { return new
Submap(from, false, KEY_NULL, true); }

  @Override
  public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE
to) { return new Submap(from, false, to, false); }

  /** A submap with given range.
   *
   * <p>This class represents a submap. One has to specify the left/right
   * limits (which can be set to -&infin; or &infin;). Since the submap is a
   * view on the map, at a given moment it could happen that the limits of
   * the range are not any longer in the main map. Thus, things such as
   * {@link java.util.SortedMap#firstKey()} or {@link java.util.Collection#size()} must be always computed
   * on-the-fly.
   */
  private
  final class Submap extends ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC implements
java.io.Serializable {
   private static final long serialVersionUID = -7046029254386353129L;

   /** The start of the submap range, unless {@link #bottom} is true. */
   KEY_GENERIC_TYPE from;
   /** The end of the submap range, unless {@link #top} is true. */
   KEY_GENERIC_TYPE to;
   /** If true, the submap range starts from -&infin;. */
```

```
 boolean bottom;
 /** If true, the submap range goes to &infin;. */
 boolean top;
 /** Cached set of entries. */
 protected transient ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> entries;
 /** Cached set of keys. */
 protected transient SORTED_SET KEY_GENERIC keys;
 /** Cached collection of values. */
 protected transient VALUE_COLLECTION VALUE_GENERIC values;

 /** Creates a new submap with given key range.
  *
  * @param from the start of the submap range.
  * @param bottom if true, the first parameter is ignored and the range starts from -&infin;.
  * @param to the
end of the submap range.
  * @param top if true, the third parameter is ignored and the range goes to &infin;.
  */
 public Submap(final KEY_GENERIC_TYPE from, final boolean bottom, final KEY_GENERIC_TYPE to, final
boolean top) {
  if (! bottom && ! top && AVL_TREE_MAP.this.compare(from, to) > 0) throw new
IllegalArgumentException("Start key (" + from  + ") is larger than end key (" + to + ")");

  this.from = from;
  this.bottom = bottom;
  this.to = to;
  this.top = top;
  this.defRetValue = AVL_TREE_MAP.this.defRetValue;
 }

 @Override
 public void clear() {
  final SubmapIterator i = new SubmapIterator();
  while(i.hasNext()) {
   i.nextEntry();
   i.remove();
  }
 }

 /** Checks whether a key is in the submap range.
  * @param k a key.
  * @return true if is the key is in the submap range.
  */
 final boolean in(final KEY_GENERIC_TYPE k) {
  return (bottom || AVL_TREE_MAP.this.compare(k, from) >= 0) &&
   (top || AVL_TREE_MAP.this.compare(k, to) < 0);
 }
```

```
  @Override
  public
 ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() {
   if (entries == null) entries = new AbstractObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC>() {
     @Override
     public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() { return new
SubmapEntryIterator(); }
     @Override
     public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator(final MAP.Entry
KEY_VALUE_GENERIC from) { return new SubmapEntryIterator(from.ENTRY_GET_KEY()); }
     @Override
     public Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator() { return
AVL_TREE_MAP.this.ENTRYSET().comparator(); }
     @Override
     SUPPRESS_WARNINGS_KEY_UNCHECKED
     public boolean contains(final Object o) {
      if (!(o instanceof Map.Entry)) return false;
      final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
      if (e.getKey() == null || ! (e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
      if (e.getValue() ==
 null || ! (e.getValue() instanceof VALUE_CLASS)) return false;
#endif
      final AVL_TREE_MAP.Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST
e.getKey()));
      return f != null && in(f.key) && e.equals(f);
     }
     @Override
     SUPPRESS_WARNINGS_KEY_UNCHECKED
     public boolean remove(final Object o) {
      if (!(o instanceof Map.Entry)) return false;
      final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
      if (e.getKey() == null || ! (e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
      if (e.getValue() == null || ! (e.getValue() instanceof VALUE_CLASS)) return false;
#endif
      final AVL_TREE_MAP.Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST
e.getKey()));
      if (f != null && in(f.key)) Submap.this.REMOVE_VALUE(f.key);
      return f != null;
     }
     @Override
     public int size() {
      int c = 0;
```

```
      for(Iterator<?> i = iterator(); i.hasNext(); i.next()) c++;
      return c;
     }
     @Override
     public
 boolean isEmpty() { return ! new SubmapIterator().hasNext(); }
     @Override
     public void clear() { Submap.this.clear(); }
     @Override
     public MAP.Entry KEY_VALUE_GENERIC first() { return firstEntry(); }
     @Override
     public MAP.Entry KEY_VALUE_GENERIC last() { return lastEntry(); }
     @Override
     public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> subSet(MAP.Entry KEY_VALUE_GENERIC
from, MAP.Entry KEY_VALUE_GENERIC to) { return subMap(from.ENTRY_GET_KEY(),
to.ENTRY_GET_KEY()).ENTRYSET(); }
     @Override
     public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> headSet(MAP.Entry KEY_VALUE_GENERIC
to) { return headMap(to.ENTRY_GET_KEY()).ENTRYSET(); }
     @Override
     public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> tailSet(MAP.Entry KEY_VALUE_GENERIC
from) { return tailMap(from.ENTRY_GET_KEY()).ENTRYSET(); }
    };

   return entries;
  }

  private class KeySet extends ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC.KeySet {
   @Override
   public
 KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new SubmapKeyIterator(); }
   @Override
   public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
SubmapKeyIterator(from); }
  }

  @Override
  public SORTED_SET KEY_GENERIC keySet() {
   if (keys == null) keys = new KeySet();
   return keys;
  }

  @Override
  public VALUE_COLLECTION VALUE_GENERIC values() {
   if (values == null) values = new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {
    @Override
    public VALUE_ITERATOR VALUE_GENERIC iterator() { return new SubmapValueIterator(); }
    @Override
```

```
    public boolean contains(final VALUE_TYPE k) { return containsValue(k); }
    @Override
    public int size() { return Submap.this.size();}
    @Override
    public void clear() { Submap.this.clear(); }
   };

  return values;
 }

 @Override
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public boolean containsKey(final KEY_TYPE k) {
  return in(KEY_GENERIC_CAST k) && AVL_TREE_MAP.this.containsKey(k);
 }

 @Override
 public
boolean containsValue(final VALUE_TYPE v) {
  final SubmapIterator i = new SubmapIterator();
  VALUE_TYPE ev;

  while(i.hasNext()) {
   ev = i.nextEntry().value;
   if (VALUE_EQUALS(ev, v)) return true;
  }

  return false;
 }

 @Override
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
  final AVL_TREE_MAP.Entry KEY_VALUE_GENERIC e;
  final KEY_GENERIC_TYPE kk = KEY_GENERIC_CAST k;
  return in(kk) && (e = findKey(kk)) != null ? e.value : this.defRetValue;
 }

 @Override
 public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
  modified = false;
  if (! in(k)) throw new IllegalArgumentException("Key (" + k + ") out of range [" + (bottom ? "-" :
String.valueOf(from)) + ", " + (top ? "-" : String.valueOf(to)) + ")");
  final VALUE_GENERIC_TYPE oldValue = AVL_TREE_MAP.this.put(k, v);
  return modified ? this.defRetValue : oldValue;
 }

 @Override
```

```
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public
VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) {
  modified = false;
  if (! in(KEY_GENERIC_CAST k)) return this.defRetValue;
  final VALUE_GENERIC_TYPE oldValue = AVL_TREE_MAP.this.REMOVE_VALUE(k);
  return modified ? oldValue : this.defRetValue;
 }

 @Override
 public int size() {
  final SubmapIterator i = new SubmapIterator();
  int n = 0;

  while(i.hasNext()) {
   n++;
   i.nextEntry();
  }

  return n;
 }

 @Override
 public boolean isEmpty() { return ! new SubmapIterator().hasNext(); }

 @Override
 public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return actualComparator; }

 @Override
 public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_TYPE to) {
  if (top) return new Submap(from, bottom, to, false);
  return compare(to, this.to) < 0 ? new Submap(from, bottom, to, false) : this;
 }

 @Override
 public SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_TYPE from) {
  if (bottom) return new Submap(from,
false, to, top);
  return compare(from, this.from) > 0 ? new Submap(from, false, to, top) : this;
 }

 @Override
 public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE
to) {
  if (top && bottom) return new Submap(from, false, to, false);
  if (! top) to = compare(to, this.to) < 0 ? to : this.to;
  if (! bottom) from = compare(from, this.from) > 0 ? from : this.from;
   if (! top && ! bottom && from == this.from && to == this.to) return this;
```

```
 return new Submap(from, false, to, false);
}

/** Locates the first entry.
 *
 * @return the first entry of this submap, or {@code null} if the submap is empty.
 */
public AVL_TREE_MAP.Entry KEY_VALUE_GENERIC firstEntry() {
 if (tree == null) return null;
 // If this submap goes to -infinity, we return the main map first entry; otherwise, we locate the start of the map.
 AVL_TREE_MAP.Entry KEY_VALUE_GENERIC e;
 if (bottom) e = firstEntry;
 else {
  e = locateKey(from);
  //
If we find either the start or something greater we're OK.
  if (compare(e.key, from) < 0) e = e.next();
 }
 // Finally, if this subset doesn't go to infinity, we check that the resulting key isn't greater than the end.
 if (e == null || ! top && compare(e.key, to) >= 0) return null;
 return e;
}

/** Locates the last entry.
 *
 * @return the last entry of this submap, or {@code null} if the submap is empty.
 */
public AVL_TREE_MAP.Entry KEY_VALUE_GENERIC lastEntry() {
 if (tree == null) return null;
 // If this submap goes to infinity, we return the main map last entry; otherwise, we locate the end of the map.
 AVL_TREE_MAP.Entry KEY_VALUE_GENERIC e;
 if (top) e = lastEntry;
 else {
  e = locateKey(to);
  // If we find something smaller than the end we're OK.
  if (compare(e.key, to) >= 0) e = e.prev();
 }
 // Finally, if this subset doesn't go to -infinity, we check that the resulting key isn't smaller than the start.
 if (e == null || !
bottom && compare(e.key, from) < 0) return null;
 return e;
}

@Override
public KEY_GENERIC_TYPE FIRST_KEY() {
 AVL_TREE_MAP.Entry KEY_VALUE_GENERIC e = firstEntry();
 if (e == null) throw new NoSuchElementException();
 return e.key;
```

```
  }

  @Override
  public KEY_GENERIC_TYPE LAST_KEY() {
   AVL_TREE_MAP.Entry KEY_VALUE_GENERIC e = lastEntry();
   if (e == null) throw new NoSuchElementException();
   return e.key;
  }

  /** An iterator for subranges.
   *
   * <p>This class inherits from {@link TreeIterator}, but overrides the methods that
   * update the pointer after a {@link java.util.ListIterator#next()} or {@link java.util.ListIterator#previous()}. If we
would
   * move out of the range of the submap we just overwrite the next or previous
   * entry with {@code null}.
   */
  private class SubmapIterator extends TreeIterator {
   SubmapIterator() { next = firstEntry(); }
   SubmapIterator(final KEY_GENERIC_TYPE k) {
    this();

    if (next != null) {
     if (! bottom
 && compare(k, next.key) < 0) prev = null;
     else if (! top && compare(k, (prev = lastEntry()).key) >= 0) next = null;
     else {
      next = locateKey(k);

      if (compare(next.key, k) <= 0) {
       prev = next;
       next = next.next();
      }
      else prev = next.prev();
     }
    }
   }

   @Override
   void updatePrevious() {
    prev = prev.prev();
    if (! bottom && prev != null && AVL_TREE_MAP.this.compare(prev.key, from) < 0) prev = null;
   }

   @Override
   void updateNext() {
    next = next.next();
    if (! top && next != null && AVL_TREE_MAP.this.compare(next.key, to) >= 0) next = null;
```

```
  }
 }

 private class SubmapEntryIterator extends SubmapIterator implements ObjectListIterator<MAP.Entry
KEY_VALUE_GENERIC>  {
  SubmapEntryIterator() {}
  SubmapEntryIterator(final KEY_GENERIC_TYPE k) { super(k); }

  @Override
  public MAP.Entry KEY_VALUE_GENERIC next() { return nextEntry(); }
  @Override
  public MAP.Entry KEY_VALUE_GENERIC previous() { return
 previousEntry(); }
 }


 /** An iterator on a subrange of keys.
  *
  * <p>This class can iterate in both directions on a subrange of the
  * keys of a threaded tree. We simply override the {@link
  * java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()} methods (and possibly their
  * type-specific counterparts) so that they return keys instead of
  * entries.
  */
 private final class SubmapKeyIterator extends SubmapIterator implements KEY_LIST_ITERATOR
KEY_GENERIC {
  public SubmapKeyIterator() { super(); }
  public SubmapKeyIterator(KEY_GENERIC_TYPE from) { super(from); }

  @Override
  public KEY_GENERIC_TYPE NEXT_KEY() { return nextEntry().key; }
  @Override
  public KEY_GENERIC_TYPE PREV_KEY() { return previousEntry().key; }
 };

 /** An iterator on a subrange of values.
  *
  * <p>This class can iterate in both directions on the values of a
  * subrange of the keys of a threaded tree. We simply override the
  * {@link java.util.ListIterator#next()}/{@link
 java.util.ListIterator#previous()} methods (and possibly their
  * type-specific counterparts) so that they return values instead of
  * entries.
  */
 private final class SubmapValueIterator extends SubmapIterator implements VALUE_LIST_ITERATOR
VALUE_GENERIC {
  @Override
  public VALUE_GENERIC_TYPE NEXT_VALUE() { return nextEntry().value; }
```

```
  @Override
  public VALUE_GENERIC_TYPE PREV_VALUE() { return previousEntry().value; }
 };
}

/** Returns a deep copy of this tree map.
 *
 * <p>This method performs a deep copy of this tree map; the data stored in the
 * set, however, is not cloned. Note that this makes a difference only for object keys.
 *
 * @return a deep copy of this tree map.
 */

@Override
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public AVL_TREE_MAP KEY_VALUE_GENERIC clone() {
 AVL_TREE_MAP KEY_VALUE_GENERIC c;
 try {
  c = (AVL_TREE_MAP KEY_VALUE_GENERIC)super.clone();
 }
 catch(CloneNotSupportedException cantHappen) {
  throw
new InternalError();
 }

 c.keys = null;
 c.values = null;
 c.entries = null;
 c.allocatePaths();

 if (count != 0) {
  // Also this apparently unfathomable code is derived from GNU libavl.
  Entry KEY_VALUE_GENERIC e, p, q, rp = new Entry KEY_VALUE_GENERIC_DIAMOND(), rq = new Entry
KEY_VALUE_GENERIC_DIAMOND();

  p = rp;
  rp.left(tree);

  q = rq;
  rq.pred(null);

  while(true) {
   if (! p.pred()) {
    e = p.left.clone();
    e.pred(q.left);
    e.succ(q);
    q.left(e);
```

```
    p = p.left;
    q = q.left;
   }
   else {
    while(p.succ()) {
     p = p.right;

     if (p == null) {
      q.right = null;
      c.tree = rq.left;

      c.firstEntry = c.tree;
      while(c.firstEntry.left != null) c.firstEntry = c.firstEntry.left;
      c.lastEntry = c.tree;
      while(c.lastEntry.right != null) c.lastEntry = c.lastEntry.right;

      return c;
     }
     q = q.right;
    }

    p = p.right;
    q = q.right;
   }

   if (! p.succ()) {
    e =
p.right.clone();
    e.succ(q.right);
    e.pred(q);
    q.right(e);
   }
  }
 }

 return c;
}


private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
 int n = count;
 EntryIterator i = new EntryIterator();
 Entry KEY_VALUE_GENERIC e;

 s.defaultWriteObject();

 while(n-- != 0) {
  e = i.nextEntry();
```

```
  s.WRITE_KEY(e.key);
  s.WRITE_VALUE(e.value);
 }
}


/** Reads the given number of entries from the input stream, returning the corresponding tree.
 *
 * @param s the input stream.
 * @param n the (positive) number of entries to read.
 * @param pred the entry containing the key that preceeds the first key in the tree.
 * @param succ the entry containing the key that follows the last key in the tree.
 */
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
private Entry KEY_VALUE_GENERIC readTree(final java.io.ObjectInputStream s, final int n, final Entry
KEY_VALUE_GENERIC pred, final Entry KEY_VALUE_GENERIC succ) throws java.io.IOException,
ClassNotFoundException
{
 if (n == 1) {
  final Entry KEY_VALUE_GENERIC top = new Entry
KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY(), VALUE_GENERIC_CAST
s.READ_VALUE());
  top.pred(pred);
  top.succ(succ);

  return top;
 }

 if (n == 2) {
  /* We handle separately this case so that recursion will
   *always* be on nonempty subtrees. */
  final Entry KEY_VALUE_GENERIC top = new Entry
KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY(), VALUE_GENERIC_CAST
s.READ_VALUE());
  top.right(new Entry KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY(),
VALUE_GENERIC_CAST s.READ_VALUE()));
  top.right.pred(top);
  top.balance(1);
  top.pred(pred);
  top.right.succ(succ);

  return top;
 }

 // The right subtree is the largest one.
 final int rightN = n / 2, leftN = n - rightN - 1;

 final Entry KEY_VALUE_GENERIC top = new Entry KEY_VALUE_GENERIC_DIAMOND();
```

```
 top.left(readTree(s, leftN, pred, top));

 top.key = KEY_GENERIC_CAST s.READ_KEY();
 top.value = VALUE_GENERIC_CAST
s.READ_VALUE();

 top.right(readTree(s, rightN, top, succ));

 if (n == (n & -n)) top.balance(1); // Quick test for determining whether n is a power of 2.

 return top;
}



private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
 s.defaultReadObject();
 /* The storedComparator is now correctly set, but we must restore
    on-the-fly the actualComparator. */
 setActualComparator();
 allocatePaths();

 if (count != 0) {
  tree = readTree(s, count, null, null);
  Entry KEY_VALUE_GENERIC e;

  e = tree;
  while(e.left() != null) e = e.left();
  firstEntry = e;

  e = tree;
  while(e.right() != null) e = e.right();
  lastEntry = e;
 }
}


#ifdef ASSERTS_CODE
 private static KEY_VALUE_GENERIC int checkTree(Entry KEY_VALUE_GENERIC e) {
  if (e == null) return 0;

  final int leftN = checkTree(e.left()), rightN = checkTree(e.right());
  if (leftN + e.balance() != rightN)
   throw new AssertionError("Mismatch
between left tree size (" + leftN + "), right tree size (" + rightN + ") and balance (" + e.balance() + ")");

  return Math.max(leftN , rightN) + 1;
```

```
 }
#endif

#ifdef TEST

 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#else
  return Integer.toBinaryString(r.nextInt());
#endif
 }

 private static VALUE_TYPE genValue() {
#if VALUE_CLASS_Byte || VALUE_CLASS_Short || VALUE_CLASS_Character
  return (VALUE_TYPE)(r.nextInt());
#elif VALUES_PRIMITIVE
  return r.NEXT_VALUE();
#elif !VALUE_CLASS_Reference || KEY_CLASS_Reference
  return Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }


 private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
 private static java.text.FieldPosition
 p = new java.text.FieldPosition(0);

 private static String format(double d) {
  StringBuffer s = new StringBuffer();
  return format.format(d, s, p).toString();
 }

 private static void speedTest(int n, boolean comp) {
  int i, j;
  AVL_TREE_MAP m;
  java.util.TreeMap t;
  KEY_TYPE k[] = new KEY_TYPE[n];
  KEY_TYPE nk[] = new KEY_TYPE[n];
  VALUE_TYPE v[] = new VALUE_TYPE[n];
  long ms;
```

```
for(i = 0; i < n; i++) {
 k[i] = genKey();
 nk[i] = genKey();
 v[i] = genValue();
}

double totPut = 0, totYes = 0, totNo = 0, totAddTo = 0, totIterFor = 0, totIterBack = 0, totRemYes = 0, d, dd, ddd;

if (comp) { for(j = 0; j < 20; j++) {

 t = new java.util.TreeMap();

 /* We first add all pairs to t. */
 for(i = 0; i < n;  i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));

 /* Then we remove the first half and put it back. */
 for(i = 0; i < n/2;  i++) t.remove(KEY2OBJ(k[i]));

 ms = System.currentTimeMillis();
 for(i = 0; i < n/2;  i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));
 d
= System.currentTimeMillis() - ms;

 /* Then we remove the other half and put it back again. */
 ms = System.currentTimeMillis();
 for(i = n/2; i < n;  i++) t.remove(KEY2OBJ(k[i]));
 dd = System.currentTimeMillis() - ms ;

 ms = System.currentTimeMillis();
 for(i = n/2; i < n;  i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));
 d += System.currentTimeMillis() - ms;
 if (j > 2) totPut += n/d;
 System.out.print("Add: " + format(n/d) +" K/s ");

 /* Then we remove again the first half. */
 ms = System.currentTimeMillis();
 for(i = 0; i < n/2;  i++) t.remove(KEY2OBJ(k[i]));
 dd += System.currentTimeMillis() - ms ;
 if (j > 2) totRemYes += n/dd;
 System.out.print("RemYes: " + format(n/dd) +" K/s ");

 /* And then we put it back. */
 for(i = 0; i < n/2;  i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));

 #if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
 /* we perform n/2 addTo() operations with get then put */
 ms = System.currentTimeMillis();
```

```
  for(i
= 0; i < n/2; i++) t.put(KEY2OBJ(k[i]), (VALUE_TYPE) ((VALUE_CLASS) t.get(KEY2OBJ(k[i])) + i));
  ddd = System.currentTimeMillis() - ms;
  if (j > 2) totAddTo += n/ddd;
  System.out.print("AddTo: " + format(n/ddd) +" K/s ");
  #endif

  /* We check for pairs in t. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) t.containsKey(KEY2OBJ(k[i]));
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totYes += d;
  System.out.print("Yes: " + format(d) +" K/s ");

  /* We check for pairs not in t. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) t.containsKey(KEY2OBJ(nk[i]));
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totNo += d;
  System.out.print("No: " + format(d) +" K/s ");

  /* We iterate on t. */
  ms = System.currentTimeMillis();
  for(Iterator it = t.entrySet().iterator(); it.hasNext(); it.next());
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totIterFor += d;
  System.out.print("IterFor:
" + format(d) +" K/s ");

  System.out.println();
  }

  System.out.println();
  System.out.println("java.util Put: " + format(totPut/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s Yes:
" + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3))+ "K/s AddTo: " + format(totAddTo/(j-3)) + " K/s
IterFor: " + format(totIterFor/(j-3))  + " K/s");

  System.out.println();

  t = null;
  totPut = totYes = totNo = totIterFor = totIterBack = totRemYes = totAddTo = 0;

  }

  for(j = 0; j < 20; j++) {

   m = new AVL_TREE_MAP();
```

```
  /* We first add all pairs to m. */
  for(i = 0; i < n;  i++) m.put(k[i], v[i]);

  /* Then we remove the first half and put it back. */
  for(i = 0; i < n/2;  i++) m.remove(k[i]);

  ms = System.currentTimeMillis();
  for(i = 0; i < n/2;  i++) m.put(k[i], v[i]);
  d = System.currentTimeMillis() - ms;

  /* Then we remove the other half and put it back again. */
  ms = System.currentTimeMillis();
  for(i = n/2; i < n;  i++) m.remove(k[i]);
  dd
= System.currentTimeMillis() - ms ;

  ms = System.currentTimeMillis();
  for(i = n/2; i < n;  i++) m.put(k[i], v[i]);
  d += System.currentTimeMillis() - ms;
  if (j > 2) totPut += n/d;
  System.out.print("Add: " + format(n/d) +" K/s ");

  /* Then we remove again the first half. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n/2;  i++) m.remove(k[i]);
  dd += System.currentTimeMillis() - ms ;
  if (j > 2) totRemYes += n/dd;
  System.out.print("RemYes: " + format(n/dd) +" K/s ");

  /* And then we put it back. */
  for(i = 0; i < n/2;  i++) m.put(k[i], v[i]);

  #if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
  /* we perform n/2 addTo() operations with get then put */
  ms = System.currentTimeMillis();
  for(i = 0; i < n/2; i++) m.addTo(k[i], (VALUE_TYPE) i);
  ddd = System.currentTimeMillis() - ms;
  if (j > 2) totAddTo += n/ddd;
  System.out.print("AddTo: " + format(n/ddd) +" K/s ");
  #endif

  /* We check for pairs in m. */
  ms = System.currentTimeMillis();
  for(i
= 0; i < n;  i++) m.containsKey(k[i]);
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totYes += d;
  System.out.print("Yes: " + format(d) +" K/s ");
```

```
  /* We check for pairs not in m. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) m.containsKey(nk[i]);
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totNo += d;
  System.out.print("No: " + format(d) +" K/s ");


  /* We iterate on m. */
  java.util.ListIterator it = (java.util.ListIterator)m.entrySet().iterator();
  ms = System.currentTimeMillis();
  for(it = (java.util.ListIterator)m.entrySet().iterator(); it.hasNext(); it.next());
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totIterFor += d;
  System.out.print("IterFor: " + format(d) +" K/s ");

  /* We iterate back on m. */
  ms = System.currentTimeMillis();
  for(; it.hasPrevious(); it.previous());
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totIterBack
 += d;
  System.out.print("IterBack: " + format(d) +" K/s ");

  System.out.println();
  }


 System.out.println();
 System.out.println("fastutil Put: " + format(totPut/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s Yes:
" + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3))+ "K/s AddTo: " + format(totAddTo/(j-3)) + " K/s
IterFor: " + format(totIterFor/(j-3))  + " K/s");

 System.out.println();

 }


 private static boolean valEquals(Object o1, Object o2) {
  return o1 == null ? o2 == null : o1.equals(o2);
 }

 private static void fatal(String msg) {
  System.out.println(msg);
  System.exit(1);
 }
```

```
 private static void ensure(boolean cond, String msg) {
  if (cond) return;
  fatal(msg);
 }

 private static void compareMT(SORTED_MAP m, SortedMap t, int level, long seed) {
  /* Now we check that m and t are equal. */
  if (!m.equals(t) || ! t.equals(m)) System.err.println("m: " + m + " t: " + t);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t)
 at start");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");



  /* Now we check that m actually holds that data. */
  for(Iterator i=t.entrySet().iterator(); i.hasNext();) {
   java.util.Map.Entry e = (java.util.Map.Entry)i.next();
   ensure(valEquals(e.getValue(), m.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after insertion (iterating on t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */
  for(Iterator i=m.entrySet().iterator(); i.hasNext();) {
   Entry e = (Entry)i.next();
   ensure(valEquals(e.getValue(), t.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after insertion (iterating on m)");
  }

  /* Now we check that m actually holds the same keys. */
  for(Iterator i=t.keySet().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(m.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a
 key ("+o+") after insertion (iterating on t)");
   ensure(m.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key ("+o+", in keySet()) after
insertion (iterating on t)");
  }

  /* Now we check that m actually holds the same keys, but iterating on m. */
  for(Iterator i=m.keySet().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(t.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key after insertion (iterating on m)");
   ensure(t.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (in keySet()) after insertion
(iterating on m)");
  }



  /* Now we check that m actually hold the same values. */
```

```
  for(Iterator i=t.values().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(m.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after insertion (iterating on
t)");
   ensure(m.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on
 a value (in values()) after insertion (iterating on t)");
  }

  /* Now we check that m actually hold the same values, but iterating on m. */
  for(Iterator i=m.values().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(t.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after insertion (iterating on
m)");
   ensure(t.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
insertion (iterating on m)");
  }

 }

 private static Object[] k, v, nk;
 private static KEY_TYPE kt[];
 private static KEY_TYPE nkt[];
 private static VALUE_TYPE vt[];
 private static AVL_TREE_MAP topMap;

 protected static void testMaps(SORTED_MAP m, SortedMap t, int n, int level) {
  long ms;
  boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement;
  Object rt = null, rm = null;

  if (level > 4) return;


  /* Now we check that both maps agree on first/last keys. */

  mThrowsNoElement = mThrowsIllegal
 = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.firstKey();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.firstKey();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): firstKey() divergence at
```

```
start in NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  if (! mThrowsNoElement) ensure(t.firstKey().equals(m.firstKey()), "Error (" + level + ", " + seed + "): m and t differ
at start on their first key (" + m.firstKey() + ", " + t.firstKey() +")");

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.lastKey();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.lastKey();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): lastKey() divergence
 at start in NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");


  if (! mThrowsNoElement) ensure(t.lastKey().equals(m.lastKey()), "Error (" + level + ", " + seed + "): m and t differ
at start on their last key (" + m.lastKey() + ", " + t.lastKey() +")");

  compareMT(m, t, level, seed);

  /* Now we check that inquiries about random data give the same answer in m and t. For
     m we use the polymorphic method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

   try {
    m.containsKey(KEY2OBJ(T));
   }
   catch (NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }

   try {
    t.containsKey(KEY2OBJ(T));
   }
   catch (NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error
 (" + level + ", " + seed + "): containsKey() divergence in NoSuchElementException (" + mThrowsNoElement + ", "
+ tThrowsNoElement + ")");
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): containsKey() divergence in
```

```
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
   if (!mThrowsNoElement && !mThrowsIllegal) {
    ensure(m.containsKey(KEY2OBJ(T)) == t.containsKey(KEY2OBJ(T)), "Error (" + level + ", " + seed + "):
divergence in keys between t and m (polymorphic method)");

#if KEY_CLASS_Object && ! (VALUES_REFERENCE)
    if ((m.GET_VALUE(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
     t.get(KEY2OBJ(T)) != null &&
     ! VALUE2OBJ(m.GET_VALUE(T)).equals(t.get(KEY2OBJ(T))))
#else
     if ((m.get(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
      t.get(KEY2OBJ(T)) != null &&
      ! m.get(KEY2OBJ(T)).equals(t.get(KEY2OBJ(T))))
#endif
      {
       System.out.println("Error
 (" + level + ", " + seed + "): divergence between t and m (polymorphic method)");
       System.exit(1);
      }
   }
  }

  /* Again, we check that inquiries about random data give the same answer in m and t, but
     for m we use the standard method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();
   mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

   try {
    m.get(KEY2OBJ(T));
   }
   catch (NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }

   try {
    t.get(KEY2OBJ(T));
   }
   catch (NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): get() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement
 + ")");
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): get() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
```

```
  if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(m.get(KEY2OBJ(T)), t.get(KEY2OBJ(T))),
"Error (" + level + ", " + seed + "): divergence between t and m (standard method)");
  }

  /* Now we put and remove random data in m and t, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();
   VALUE_TYPE U = genValue();

   mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

   try {
    rm = m.put(KEY2OBJ(T), VALUE2OBJ(U));
   }
   catch (NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }

   try {
    rt = t.put(KEY2OBJ(T), VALUE2OBJ(U));
   }
   catch (NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException
 e) { tThrowsIllegal = true; }


   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): put() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): put() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
   if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(rm, rt), "Error (" + level + ", " + seed + "):
divergence in put() between t and m (" + rt + ", " + rm + ")");

   T = genKey();


   mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

   try {
    rm = m.remove(KEY2OBJ(T));
   }
   catch (NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }

   try {
    rt = t.remove(KEY2OBJ(T));
   }
   catch (NoSuchElementException e) { tThrowsNoElement = true; }
```

```
  catch
 (IllegalArgumentException e) { tThrowsIllegal = true; }


  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(rm, rt), "Error (" + level + ", " + seed + "):
divergence in remove() between t and m (" + rt + ", " + rm + ")");
  }

 compareMT(m, t, level, seed);

 #if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
 /* Now we check that addTo results in the same values as get/put */
 if (m instanceof AVL_TREE_MAP) {
  for (Iterator i=m.entrySet().iterator(); i.hasNext();) {
   Entry e = (Entry)i.next();
   VALUE_TYPE incr = genValue();
   ((AVL_TREE_MAP) m).addTo(e.key, incr);
   t.put(e.key,
 (VALUE_TYPE) ((VALUE_CLASS) t.get(e.key) + incr));
  }
  compareMT(m, t, level, seed);

  /* Now we make sure that addTo works with a key that did not previously exist (with special default return value)*/
  KEY_TYPE newKey;

  do {
   newKey = genKey();
  } while (m.containsKey(newKey));

  VALUE_TYPE newValue, drv;
  drv = m.defaultReturnValue();
  newValue = genValue();
  m.defaultReturnValue(genValue());

  ((AVL_TREE_MAP) m).addTo(newKey, newValue);
  t.put(newKey, (VALUE_TYPE) (newValue + m.defaultReturnValue()));

  compareMT(m, t, level, seed);
  m.defaultReturnValue(drv); // set the default value back to the old drv
 }
 #endif

 /* Now we check that both maps agree on first/last keys. */
```

```
  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.firstKey();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.firstKey();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true;
 }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): firstKey() divergence in
NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  if (! mThrowsNoElement) ensure(t.firstKey().equals(m.firstKey()), "Error (" + level + ", " + seed + "): m and t differ
on their first key (" + m.firstKey() + ", " + t.firstKey() +")");

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.lastKey();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.lastKey();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): lastKey() divergence in
NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");

  if (! mThrowsNoElement) ensure(t.lastKey().equals(m.lastKey()), "Error (" + level + ", " + seed + "): m and t differ
on their last key (" + m.lastKey() + ",
 " + t.lastKey() +")");

  /* Now we check cloning. */

  if (level == 0) {
   ensure(m.equals(((AVL_TREE_MAP)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
   ensure(((AVL_TREE_MAP)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
  }

  int h = m.hashCode();


  /* Now we save and read m. */

  SORTED_MAP m2 = null;
```

```
  try {
   java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
   java.io.OutputStream os = new java.io.FileOutputStream(ff);
   java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

   oos.writeObject(m);
   oos.close();

   java.io.InputStream is = new java.io.FileInputStream(ff);
   java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

   m2 = (SORTED_MAP)ois.readObject();
   ois.close();
   ff.delete();
  }
  catch(Exception e) {
   e.printStackTrace();
   System.exit(1);
  }

#if !VALUE_CLASS_Reference
  ensure(m2.hashCode()
 == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

  /* Now we check that m2 actually holds that data. */

  ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
  ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
#else
  m2.clear();
  m2.putAll(m);
#endif

  /* Now we take out of m everything, and check that it is empty. */

  for(Iterator i=t.keySet().iterator(); i.hasNext();) m2.remove(i.next());

  ensure(m2.isEmpty(), "Error (" + level + ", " + seed + "): m2 is not empty (as it should be)");

  /* Now we play with iterators. */

  {
   java.util.ListIterator i, j;
   Map.Entry E, F;
   Object J;
   i = (java.util.ListIterator)m.entrySet().iterator();
   j = new java.util.LinkedList(t.entrySet()).listIterator();
```

```
  for(int k = 0; k < 2*n; k++) {
   ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext()");
   ensure(i.hasPrevious()
 == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious()");

   if (r.nextFloat() < .8 && i.hasNext()) {
    ensure((E=(Entry)i.next()).getKey().equals(J = (F=(Map.Entry)j.next()).getKey()), "Error (" + level + ", " + seed +
"): divergence in next()");

    if (r.nextFloat() < 0.3) {
     i.remove();
     j.remove();
     t.remove(J);
    }
    else if (r.nextFloat() < 0.3) {
     Object U = VALUE2OBJ(genValue());
     E.setValue(U);
              t.put(F.getKey(), U);
    }
   }
   else if (r.nextFloat() < .2 && i.hasPrevious()) {
    ensure((E=(Entry)i.previous()).getKey().equals(J = (F=(Map.Entry)j.previous()).getKey()), "Error (" + level + ", " +
seed + "): divergence in previous()");

    if (r.nextFloat() < 0.3) {
     i.remove();
     j.remove();
     t.remove(J);
    }
    else if (r.nextFloat() < 0.3) {
     Object U = VALUE2OBJ(genValue());
     E.setValue(U);
              t.put(F.getKey(), U);
    }
   }

   ensure(i.nextIndex()
 == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
   ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

  }

 }



 {
  boolean badPrevious = false;
```

```
  Object previous = null;
  it.unimi.dsi.fastutil.BidirectionalIterator i;
  java.util.ListIterator j;
  Object I, J;
  KEY_TYPE from = genKey();
  j = new java.util.LinkedList(t.keySet()).listIterator();
  while(j.hasNext()) {
   Object k = j.next();
   if (((Comparable)k).compareTo(KEY2OBJ(from)) > 0) {
    badPrevious = true;
    j.previous();
    break;
   }
   previous = k;
  }

  i = (it.unimi.dsi.fastutil.BidirectionalIterator)((SORTED_SET)m.keySet()).iterator(from);

  for(int k = 0; k < 2*n; k++) {
   ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting
point " + from + ")");
   ensure(i.hasPrevious()
 == j.hasPrevious() || badPrevious && (i.hasPrevious() == (previous != null)), "Error (" + level + ", " + seed + "):
divergence in hasPrevious() (iterator with starting point " + from + ")" + badPrevious);

   if (r.nextFloat() < .8 && i.hasNext()) {
    ensure((I = i.next()).equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next() (" + I + ", " + J + ",
iterator with starting point " + from + ")");
    //System.err.println("Done next " + I + " " + J + "  " + badPrevious);

    badPrevious = false;

    if (r.nextFloat() < 0.5) {
     //System.err.println("Removing in next");
     i.remove();
     j.remove();
     t.remove(J);
    }
   }
   else if (!badPrevious && r.nextFloat() < .2 && i.hasPrevious()) {
    ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

    if (r.nextFloat() < 0.5) {
     //System.err.println("Removing
 in prev");
     i.remove();
     j.remove();
```

```
    t.remove(J);
   }
  }
 }

}

/* Now we check that m actually holds that data. */

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after iteration");

/* Now we select a pair of keys and create a submap. */

if (! m.isEmpty()) {
 java.util.ListIterator i;
 Object start = m.firstKey(), end = m.firstKey();
 for(i = (java.util.ListIterator)m.keySet().iterator(); i.hasNext() && r.nextFloat() < .3; start = end = i.next());
 for(; i.hasNext() && r.nextFloat() < .95; end = i.next());

 //System.err.println("Checking subMap from " + start + " to " + end + " (level=" + (level+1) + ")...");
 testMaps((SORTED_MAP)m.subMap((KEY_CLASS)start, (KEY_CLASS)end), t.subMap(start, end), n, level + 1);

 ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after subMap");
 ensure(t.equals(m),
"Error (" + level + ", " + seed + "): ! t.equals(m) after subMap");

 //System.err.println("Checking headMap to " + end + " (level=" + (level+1) + ")...");
 testMaps((SORTED_MAP)m.headMap((KEY_CLASS)end), t.headMap(end), n, level + 1);

 ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after headMap");
 ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after headMap");

 //System.err.println("Checking tailMap from " + start + " (level=" + (level+1) + ")...");
 testMaps((SORTED_MAP)m.tailMap((KEY_CLASS)start), t.tailMap(start), n, level + 1);

 ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after tailMap");
 ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after tailMap");
}


}


private static void runTest(int n) {
 AVL_TREE_MAP m = new AVL_TREE_MAP();
 SortedMap t = new java.util.TreeMap();
 topMap = m;
```

```
  k = new Object[n];
  v = new Object[n];
  nk = new
 Object[n];
  kt = new KEY_TYPE[n];
  nkt = new KEY_TYPE[n];
  vt = new VALUE_TYPE[n];

  for(int i = 0; i < n; i++) {
#if KEY_CLASS_Object
   k[i] = kt[i] = genKey();
   nk[i] = nkt[i] = genKey();
#else
   k[i] = new KEY_CLASS(kt[i] = genKey());
   nk[i] = new KEY_CLASS(nkt[i] = genKey());
#endif
#if VALUES_REFERENCE
   v[i] = vt[i] = genValue();
#else
   v[i] = new VALUE_CLASS(vt[i] = genValue());
#endif
  }

  /* We add pairs to t. */
  for(int i = 0; i < n;  i++) t.put(k[i], v[i]);

  /* We add to m the same data */
  m.putAll(t);

  testMaps(m, t, n, 0);

  System.out.println("Test OK");
  return;
 }


 public static void main(String args[]) {
  int n  = Integer.parseInt(args[1]);
  if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

  try {
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
   else if ("test".equals(args[0])) runTest(n);
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed:
 " + seed);
  }
```

```
 }

#endif


}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AVLTreeMap.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import it.unimi.dsi.fastutil.IndirectPriorityQueue;

/** A type-specific {@link IndirectPriorityQueue}.
*
* <p>Additionally, this interface strengthens {@link #comparator()}.
*/

public interface INDIRECT_PRIORITY_QUEUE extends IndirectPriorityQueue<KEY_CLASS> {

 /** Returns the type-specific
 comparator associated with this queue.
  *
  * <p>Note that this specification strengthens the one given in {@link IndirectPriorityQueue}.
  *
  * @return the comparator associated with this queue.
  * @see IndirectPriorityQueue#comparator()
  */
```

```
 @Override
 KEY_COMPARATOR comparator();
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/IndirectPriorityQueue.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/




 /** Returns the element of the given big array of specified index.
  *
  * @param array a big array.
  * @param index a position in the big array.
  * @return the element of the big array at the specified position.
  */
 public static KEY_GENERIC KEY_GENERIC_TYPE get(final KEY_GENERIC_TYPE[][] array, final long index)
 {
  return array[segment(index)][displacement(index)];
 }

 /** Sets the element of the given big array of specified index.
  *
  * @param array a big array.
  * @param index a position in the big array.
  * @param value the new value for the array element at the specified position.
  */
 public static KEY_GENERIC void set(final KEY_GENERIC_TYPE[][] array, final long index,
KEY_GENERIC_TYPE value) {
  array[segment(index)][displacement(index)] = value;
```

```
 }

#if KEY_CLASS_Long || KEY_CLASS_Integer
 /** Returns the length of the given big atomic array.
  *
  * @param array a big atomic array.
  * @return the length of the given big atomic array.
  */
 public static long length(final ATOMIC_ARRAY[] array) {
  final int length = array.length;
  return length == 0 ? 0 : start(length - 1) + array[length - 1].length();
 }

 /** Returns the element of the given big atomic array of specified index.
  *
  * @param array a big atomic array.
  * @param index a position in the big atomic array.

 * @return the element of the big atomic array at the specified position.
  */
 public static KEY_TYPE get(final ATOMIC_ARRAY[] array, final long index) {
  return array[segment(index)].get(displacement(index));
 }

 /** Sets an element of the given big atomic array to a specified value
  *
  * @param array a big atomic array.
  * @param index a position in the big atomic array.
  * @param value a new value for the element of the big atomic array at the specified position.
  */
 public static void set(final ATOMIC_ARRAY[] array, final long index, KEY_TYPE value) {
  array[segment(index)].set(displacement(index), value);
 }

 /** Atomically sets an element of the given big atomic array to a specified value, returning the old value.
  *
  * @param array a big atomic array.
  * @param index a position in the big atomic array.
  * @param value a new value for the element of the big atomic array at the specified position.
  * @return the old value of the element of the big atomic array at
 the specified position.
  */
 public static KEY_TYPE getAndSet(final ATOMIC_ARRAY[] array, final long index, KEY_TYPE value) {
  return array[segment(index)].getAndSet(displacement(index), value);
 }

 /** Atomically adds a value to an element of the given big atomic array, returning the old value.
  *
```

```
 * @param array a big atomic array.
 * @param index a position in the big atomic array.
 * @param value a value to add to the element of the big atomic array at the specified position.
 * @return the old value of the element of the big atomic array at the specified position.
 */
public static KEY_TYPE getAndAdd(final ATOMIC_ARRAY[] array, final long index, KEY_TYPE value) {
 return array[segment(index)].getAndAdd(displacement(index), value);
}

/** Atomically adds a value to an element of the given big atomic array, returning the new value.
 *
 * @param array a big atomic array.
 * @param index a position in the big atomic array.
 * @param value a value to add to the element
of the big atomic array at the specified position.
 * @return the new value of the element of the big atomic array at the specified position.
 */
public static KEY_TYPE addAndGet(final ATOMIC_ARRAY[] array, final long index, KEY_TYPE value) {
 return array[segment(index)].addAndGet(displacement(index), value);
}

/** Atomically increments an element of the given big atomic array, returning the old value.
 *
 * @param array a big atomic array.
 * @param index a position in the big atomic array.
 * @return the old value of the element of the big atomic array at the specified position.
 */
public static KEY_TYPE getAndIncrement(final ATOMIC_ARRAY[] array, final long index) {
 return array[segment(index)].getAndDecrement(displacement(index));
}

/** Atomically increments an element of the given big atomic array, returning the new value.
 *
 * @param array a big atomic array.
 * @param index a position in the big atomic array.
 * @return the new value of the element of the
big atomic array at the specified position.
 */
public static KEY_TYPE incrementAndGet(final ATOMIC_ARRAY[] array, final long index) {
 return array[segment(index)].incrementAndGet(displacement(index));
}

/** Atomically decrements an element of the given big atomic array, returning the old value.
 *
 * @param array a big atomic array.
 * @param index a position in the big atomic array.
 * @return the old value of the element of the big atomic array at the specified position.
 */
```

```
 public static KEY_TYPE getAndDecrement(final ATOMIC_ARRAY[] array, final long index) {
  return array[segment(index)].getAndDecrement(displacement(index));
 }

 /** Atomically decrements an element of the given big atomic array, returning the new value.
  *
  * @param array a big atomic array.
  * @param index a position in the big atomic array.
  * @return the new value of the element of the big atomic array at the specified position.
  */
 public static KEY_TYPE decrementAndGet(final ATOMIC_ARRAY[]
 array, final long index) {
  return array[segment(index)].decrementAndGet(displacement(index));
 }

 /** Atomically sets an element of the given big atomic array of specified index to specified value, given
  *  the current value is equal to a given expected value.
  *
  * @param array a big atomic array.
  * @param index a position in the big atomic array.
  * @param expected an expected value for the element of the big atomic array at the specified position.
  * @param value a new value for the element of the big atomic array at the specified position.
  * @return the element of the big atomic array at the specified position.
  */
 public static boolean compareAndSet(final ATOMIC_ARRAY[] array, final long index, KEY_TYPE expected,
KEY_TYPE value) {
  return array[segment(index)].compareAndSet(displacement(index), expected, value);
 }

#endif

 /** Swaps the element of the given big array of specified indices.
  *
  * @param array a big array.
  * @param first a position in the
 big array.
  * @param second a position in the big array.
  */
 public static KEY_GENERIC void swap(final KEY_GENERIC_TYPE[][] array, final long first, final long second)
{
  final KEY_GENERIC_TYPE t = array[segment(first)][displacement(first)];
  array[segment(first)][displacement(first)] = array[segment(second)][displacement(second)];
  array[segment(second)][displacement(second)] = t;
 }

 /** Reverses the order of the elements in the specified big array.
  *
  * @param a the big array to be reversed.
```

```
 * @return {@code a}.
 */
public static KEY_GENERIC KEY_GENERIC_TYPE[][] reverse(final KEY_GENERIC_TYPE[][] a) {
 final long length = length(a);
 for(long i = length / 2; i-- != 0;) swap(a, i, length - i- 1);
 return a;
}

#if KEYS_PRIMITIVE && ! KEY_CLASS_Boolean
/** Adds the specified increment the element of the given big array of specified index.
 *
 * @param array a big array.
 * @param index a position in the big array.
 * @param incr the increment
 */
public static
void add(final KEY_GENERIC_TYPE[][] array, final long index, KEY_GENERIC_TYPE incr) {
 array[segment(index)][displacement(index)] += incr;
}

/** Multiplies by the specified factor the element of the given big array of specified index.
 *
 * @param array a big array.
 * @param index a position in the big array.
 * @param factor the factor
 */
public static void mul(final KEY_GENERIC_TYPE[][] array, final long index, KEY_GENERIC_TYPE factor) {
 array[segment(index)][displacement(index)] *= factor;
}

/** Increments the element of the given big array of specified index.
 *
 * @param array a big array.
 * @param index a position in the big array.
 */
public static void incr(final KEY_GENERIC_TYPE[][] array, final long index) {
 array[segment(index)][displacement(index)]++;
}

/** Decrements the element of the given big array of specified index.
 *
 * @param array a big array.
 * @param index a position in the big array.
 */
public static void decr(final KEY_GENERIC_TYPE[][]
array, final long index) {
 array[segment(index)][displacement(index)]--;
}
```

```
#endif


 /** Returns the length of the given big array.
  *
  * @param array a big array.
  * @return the length of the given big array.
  */
 public static KEY_GENERIC long length(final KEY_GENERIC_TYPE[][] array) {
  final int length = array.length;
  return length == 0 ? 0 : start(length - 1) + array[length - 1].length;
 }

 /** Copies a big array from the specified source big array, beginning at the specified position, to the specified
position of the destination big array.
  * Handles correctly overlapping regions of the same big array.
  *
  * @param srcArray the source big array.
  * @param srcPos the starting position in the source big array.
  * @param destArray the destination big array.
  * @param destPos the starting position in the destination data.
  * @param length the number of elements to be copied.
  */
 public static KEY_GENERIC void copy(final KEY_GENERIC_TYPE[][] srcArray,
 final long srcPos, final KEY_GENERIC_TYPE[][] destArray, final long destPos, long length) {
  if (destPos <= srcPos) {
   int srcSegment = segment(srcPos);
   int destSegment = segment(destPos);
   int srcDispl = displacement(srcPos);
   int destDispl = displacement(destPos);
   int l;
   while(length > 0) {
    l = (int)Math.min(length, Math.min(srcArray[srcSegment].length - srcDispl, destArray[destSegment].length -
destDispl));
    if (l == 0) throw new ArrayIndexOutOfBoundsException();
    System.arraycopy(srcArray[srcSegment], srcDispl, destArray[destSegment], destDispl, l);
    if ((srcDispl += l) == SEGMENT_SIZE) {
     srcDispl = 0;
     srcSegment++;
    }
    if ((destDispl += l) == SEGMENT_SIZE) {
     destDispl = 0;
     destSegment++;
    }
    length -= l;
   }
```

```
 }
 else {
  int srcSegment = segment(srcPos + length);
  int destSegment = segment(destPos + length);
  int srcDispl = displacement(srcPos + length);
  int destDispl = displacement(destPos + length);
  int l;
  while(length
> 0) {
   if (srcDispl == 0) {
    srcDispl = SEGMENT_SIZE;
    srcSegment--;
   }
   if (destDispl == 0) {
    destDispl = SEGMENT_SIZE;
    destSegment--;
   }
   l = (int)Math.min(length, Math.min(srcDispl, destDispl));
   if (l == 0) throw new ArrayIndexOutOfBoundsException();
   System.arraycopy(srcArray[srcSegment], srcDispl - l, destArray[destSegment], destDispl - l, l);
   srcDispl -= l;
   destDispl -= l;
   length -= l;
  }
 }
}

/** Copies a big array from the specified source big array, beginning at the specified position, to the specified
position of the destination array.
 *
 * @param srcArray the source big array.
 * @param srcPos the starting position in the source big array.
 * @param destArray the destination array.
 * @param destPos the starting position in the destination data.
 * @param length the number of elements to be copied.
 */
public static KEY_GENERIC void copyFromBig(final KEY_GENERIC_TYPE[][] srcArray, final long
srcPos, final KEY_GENERIC_TYPE[] destArray, int destPos, int length) {
 int srcSegment = segment(srcPos);
 int srcDispl = displacement(srcPos);
 int l;
 while(length > 0) {
  l = Math.min(srcArray[srcSegment].length - srcDispl, length);
  if (l == 0) throw new ArrayIndexOutOfBoundsException();
  System.arraycopy(srcArray[srcSegment], srcDispl, destArray, destPos, l);
  if ((srcDispl += l) == SEGMENT_SIZE) {
   srcDispl = 0;
   srcSegment++;
```

```
  }
  destPos += l;
  length -= l;
 }
}

/** Copies an array from the specified source array, beginning at the specified position, to the specified position of
the destination big array.
 *
 * @param srcArray the source array.
 * @param srcPos the starting position in the source array.
 * @param destArray the destination big array.
 * @param destPos the starting position in the destination data.
 * @param length the number of elements to be copied.
 */
public static KEY_GENERIC void copyToBig(final KEY_GENERIC_TYPE[] srcArray,
int srcPos, final KEY_GENERIC_TYPE[][] destArray, final long destPos, long length) {
 int destSegment = segment(destPos);
 int destDispl = displacement(destPos);
 int l;
 while(length > 0) {
  l = (int)Math.min(destArray[destSegment].length - destDispl, length);
  if (l == 0) throw new ArrayIndexOutOfBoundsException();
  System.arraycopy(srcArray, srcPos, destArray[destSegment], destDispl, l);
  if ((destDispl += l) == SEGMENT_SIZE) {
   destDispl = 0;
   destSegment++;
  }
  srcPos += l;
  length -= l;
 }
}

#if KEY_CLASS_Object
/** Turns a standard array into a big array.
 *
 * <p>Note that the returned big array might contain as a segment the original array.
 *
 * @param array an array.
 * @return a new big array with the same length and content of {@code array}.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static <K> K[][] wrap(final K[] array) {
 if (array.length == 0 && array.getClass() == Object[].class) return KEY_GENERIC_BIG_ARRAY_CAST
BIG_ARRAYS.EMPTY_BIG_ARRAY;
 if
```

```
 (array.length <= SEGMENT_SIZE) {
   final K[][] bigArray = (K[][])java.lang.reflect.Array.newInstance(array.getClass(), 1);
   bigArray[0] = array;
   return bigArray;
  }
  final K[][] bigArray = (K[][])BIG_ARRAYS.newBigArray(array.getClass(), array.length);
  for(int i = 0; i < bigArray.length; i++) System.arraycopy(array, (int)start(i), bigArray[i], 0, bigArray[i].length);
  return bigArray;
 }

#else
 /** Turns a standard array into a big array.
  *
  * <p>Note that the returned big array might contain as a segment the original array.
  *
  * @param array an array.
  * @return a new big array with the same length and content of {@code array}.
  */

 public static KEY_TYPE[][] wrap(final KEY_TYPE[] array) {
  if (array.length == 0) return BIG_ARRAYS.EMPTY_BIG_ARRAY;
  if (array.length <= SEGMENT_SIZE) return new KEY_TYPE[][] { array };
  final KEY_TYPE[][] bigArray = BIG_ARRAYS.newBigArray(array.length);
  for(int i = 0; i < bigArray.length; i++) System.arraycopy(array,
 (int)start(i), bigArray[i], 0, bigArray[i].length);
  return bigArray;
 }
#endif
 /** Ensures that a big array can contain the given number of entries.
  *
  * <p>If you cannot foresee whether this big array will need again to be
  * enlarged, you should probably use {@code grow()} instead.
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new minimum length for this big array.
  * @return {@code array}, if it contains {@code length} entries or more; otherwise,
  * a big array with {@code length} entries whose first {@code length(array)}
  * entries are the same as those of {@code array}.
  */
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] ensureCapacity(final KEY_GENERIC_TYPE[][] array,
final long length) {
  return ensureCapacity(array, length, length(array));
 }

#if KEY_CLASS_Object
```

```
/** Forces
 a big array to contain the given number of entries, preserving just a part of the big array.
  *
  * <p>This method returns a new big array of the given length whose element
  * are of the same class as of those of {@code array}.
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new minimum length for this big array.
  * @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.
  * @return a big array with {@code length} entries whose first {@code preserve}
  * entries are the same as those of {@code array}.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] forceCapacity(final KEY_GENERIC_TYPE[][] array,
final long length, final long preserve) {
  ensureLength(length);
  final int valid = array.length - (array.length
 == 0 || array.length > 0 && array[array.length - 1].length == SEGMENT_SIZE ? 0 : 1);
  final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
  final KEY_GENERIC_TYPE[][] base = java.util.Arrays.copyOf(array, baseLength);
  final Class<?> componentType = array.getClass().getComponentType();
  final int residual = (int)(length & SEGMENT_MASK);
  if (residual != 0) {
   for(int i = valid; i < baseLength - 1; i++) base[i] =
(KEY_GENERIC_TYPE[])java.lang.reflect.Array.newInstance(componentType.getComponentType(),
SEGMENT_SIZE);
   base[baseLength - 1] =
(KEY_GENERIC_TYPE[])java.lang.reflect.Array.newInstance(componentType.getComponentType(), residual);
  }
  else for(int i = valid; i < baseLength; i++) base[i] =
(KEY_GENERIC_TYPE[])java.lang.reflect.Array.newInstance(componentType.getComponentType(),
SEGMENT_SIZE);

  if (preserve - (valid * (long)SEGMENT_SIZE) > 0) copy(array, valid * (long)SEGMENT_SIZE, base, valid *
(long)SEGMENT_SIZE, preserve - (valid * (long)SEGMENT_SIZE));
  return
 base;
 }

 /** Ensures that a big array can contain the given number of entries, preserving just a part of the big array.
  *
  * <p>This method returns a new big array of the given length whose element
  * are of the same class as of those of {@code array}.
```

```
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new minimum length for this big array.
  * @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.
  * @return {@code array}, if it can contain {@code length} entries or more; otherwise,
  * a big array with {@code length} entries whose first {@code preserve}
  * entries are the same as those of {@code array}.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] ensureCapacity(final KEY_GENERIC_TYPE[][]
 array, final long length, final long preserve) {
  return length > length(array) ? forceCapacity(array, length, preserve) : array;
 }

#else

 /** Forces a big array to contain the given number of entries, preserving just a part of the big array.
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new minimum length for this big array.
  * @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.
  * @return a big array with {@code length} entries whose first {@code preserve}
  * entries are the same as those of {@code array}.
  */
 public static KEY_TYPE[][] forceCapacity(final KEY_TYPE[][] array, final long length, final long preserve) {
  ensureLength(length);
  final int valid = array.length - (array.length == 0 || array.length
 > 0 && array[array.length - 1].length == SEGMENT_SIZE ? 0 : 1);
  final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
  final KEY_TYPE[][] base = java.util.Arrays.copyOf(array, baseLength);
  final int residual = (int)(length & SEGMENT_MASK);
  if (residual != 0) {
   for(int i = valid; i < baseLength - 1; i++) base[i] = new KEY_TYPE[SEGMENT_SIZE];
   base[baseLength - 1] = new KEY_TYPE[residual];
  }
  else for(int i = valid; i < baseLength; i++) base[i] = new KEY_TYPE[SEGMENT_SIZE];

  if (preserve - (valid * (long)SEGMENT_SIZE) > 0) copy(array, valid * (long)SEGMENT_SIZE, base, valid *
(long)SEGMENT_SIZE, preserve - (valid * (long)SEGMENT_SIZE));
  return base;
```

```
 }

 /** Ensures that a big array can contain the given number of entries, preserving just a part of the big array.
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param
 array a big array.
  * @param length the new minimum length for this big array.
  * @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.
  * @return {@code array}, if it can contain {@code length} entries or more; otherwise,
  * a big array with {@code length} entries whose first {@code preserve}
  * entries are the same as those of {@code array}.
  */
 public static KEY_TYPE[][] ensureCapacity(final KEY_TYPE[][] array, final long length, final long preserve) {
  return length > length(array) ? forceCapacity(array, length, preserve) : array;
 }

#endif

 /** Grows the given big array to the maximum between the given length and
  * the current length increased by 50%, provided that the given
  * length is larger than the current length.
  *
  * <p>If you want complete control on the big array growth, you
  * should probably use {@code ensureCapacity()} instead.
  *
  * <p><strong>Warning:</strong> the returned array might
 use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new minimum length for this big array.
  * @return {@code array}, if it can contain {@code length}
  * entries; otherwise, a big array with
  * max({@code length},{@code length(array)}/&phi;) entries whose first
  * {@code length(array)} entries are the same as those of {@code array}.
  * */

 public static KEY_GENERIC KEY_GENERIC_TYPE[][] grow(final KEY_GENERIC_TYPE[][] array, final long
length) {
  final long oldLength = length(array);
  return length > oldLength ? grow(array, length, oldLength) : array;
 }

 /** Grows the given big array to the maximum between the given length and
```

```
  * the current length increased by 50%, provided that the given
  * length is larger than the current length, preserving just a part of the big array.
  *
  * <p>If you want complete control on the big array growth, you
  * should probably
 use {@code ensureCapacity()} instead.
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new minimum length for this big array.
  * @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.
  * @return {@code array}, if it can contain {@code length}
  * entries; otherwise, a big array with
  * max({@code length},{@code length(array)}/&phi;) entries whose first
  * {@code preserve} entries are the same as those of {@code array}.
  * */

 public static KEY_GENERIC KEY_GENERIC_TYPE[][] grow(final KEY_GENERIC_TYPE[][] array, final long
length, final long preserve) {
  final long oldLength = length(array);
  return length > oldLength ? ensureCapacity(array, Math.max(oldLength + (oldLength >> 1), length), preserve) :
array;
 }

#if KEY_CLASS_Object

 /**
 Trims the given big array to the given length.
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new maximum length for the big array.
  * @return {@code array}, if it contains {@code length}
  * entries or less; otherwise, a big array with
  * {@code length} entries whose entries are the same as
  * the first {@code length} entries of {@code array}.
  *
  */

 public static KEY_GENERIC KEY_GENERIC_TYPE[][] trim(final KEY_GENERIC_TYPE[][] array, final long
length) {
  ensureLength(length);
  final long oldLength = length(array);
```

```
  if (length >= oldLength) return array;
  final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
  final KEY_GENERIC_TYPE[][] base = java.util.Arrays.copyOf(array, baseLength);
  final int residual = (int)(length & SEGMENT_MASK);
  if (residual != 0) base[baseLength
 - 1] = ARRAYS.trim(base[baseLength - 1], residual);
  return base;
 }

#else

 /** Trims the given big array to the given length.
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new maximum length for the big array.
  * @return {@code array}, if it contains {@code length}
  * entries or less; otherwise, a big array with
  * {@code length} entries whose entries are the same as
  * the first {@code length} entries of {@code array}.
  *
  */

 public static KEY_GENERIC KEY_GENERIC_TYPE[][] trim(final KEY_GENERIC_TYPE[][] array, final long
length) {
  ensureLength(length);
  final long oldLength = length(array);
  if (length >= oldLength) return array;
  final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
  final KEY_TYPE[][] base = java.util.Arrays.copyOf(array, baseLength);
  final
 int residual = (int)(length & SEGMENT_MASK);
  if (residual != 0) base[baseLength - 1] = ARRAYS.trim(base[baseLength - 1], residual);
  return base;
 }

#endif

 /** Sets the length of the given big array.
  *
  * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
  * array, which must be considered read-only after calling this method.
  *
  * @param array a big array.
  * @param length the new length for the big array.
  * @return {@code array}, if it contains exactly {@code length}
```

```
 * entries; otherwise, if it contains <em>more</em> than
 * {@code length} entries, a big array with {@code length} entries
 * whose entries are the same as the first {@code length} entries of
 * {@code array}; otherwise, a big array with {@code length} entries
 * whose first {@code length(array)} entries are the same as those of
 * {@code array}.
 *
 */

 public static KEY_GENERIC KEY_GENERIC_TYPE[][] setLength(final KEY_GENERIC_TYPE[][] array, final long
 length) {
  final long oldLength = length(array);
  if (length == oldLength) return array;
  if (length < oldLength) return trim(array, length);
  return ensureCapacity(array, length);
 }

 /** Returns a copy of a portion of a big array.
  *
  * @param array a big array.
  * @param offset the first element to copy.
  * @param length the number of elements to copy.
  * @return a new big array containing {@code length} elements of {@code array} starting at {@code offset}.
  */

 public static KEY_GENERIC KEY_GENERIC_TYPE[][] copy(final KEY_GENERIC_TYPE[][] array, final long offset, final long length) {
  ensureOffsetLength(array, offset, length);
  final KEY_GENERIC_TYPE[][] a =
#if KEY_CLASS_Object
   BIG_ARRAYS.newBigArray(array, length);
#else
   BIG_ARRAYS.newBigArray(length);
#endif
  copy(array, offset, a, 0, length);
  return a;
 }

 /** Returns a copy of a big array.
  *
  * @param array a big array.
  * @return a copy of {@code array}.
  */

 public static KEY_GENERIC KEY_GENERIC_TYPE[][]
 copy(final KEY_GENERIC_TYPE[][] array) {
  final KEY_GENERIC_TYPE[][] base = array.clone();
```

```
 for(int i = base.length; i-- != 0;) base[i] = array[i].clone();
 return base;
}

/** Fills the given big array with the given value.
 *
 * <p>This method uses a backward loop. It is significantly faster than the corresponding
 * method in {@link java.util.Arrays}.
 *
 * @param array a big array.
 * @param value the new value for all elements of the big array.
 */

public static KEY_GENERIC void fill(final KEY_GENERIC_TYPE[][] array, final KEY_GENERIC_TYPE value)
{
 for(int i = array.length; i-- != 0;) java.util.Arrays.fill(array[i], value);
}

/** Fills a portion of the given big array with the given value.
 *
 * <p>If possible (i.e., {@code from} is 0) this method uses a
 * backward loop. In this case, it is significantly faster than the
 * corresponding method in {@link java.util.Arrays}.
 *
 * @param array a big array.
 * @param from the starting index of
the portion to fill.
 * @param to the end index of the portion to fill.
 * @param value the new value for all elements of the specified portion of the big array.
 */

public static KEY_GENERIC void fill(final KEY_GENERIC_TYPE[][] array, final long from, long to, final
KEY_GENERIC_TYPE value) {
 final long length = length(array);
 BigArrays.ensureFromTo(length, from, to);
 if (length == 0) return; // To avoid addressing array[0]
 int fromSegment = segment(from);
 int toSegment = segment(to);
 int fromDispl = displacement(from);
 int toDispl = displacement(to);
 if (fromSegment == toSegment) {
  java.util.Arrays.fill(array[fromSegment], fromDispl, toDispl, value);
  return;
 }

 if (toDispl != 0) java.util.Arrays.fill(array[toSegment], 0, toDispl, value);
 while(--toSegment > fromSegment) java.util.Arrays.fill(array[toSegment], value);
 java.util.Arrays.fill(array[fromSegment], fromDispl, SEGMENT_SIZE, value);
```

```
 }


 /** Returns true if the two big arrays are elementwise
 equal.
  *
  * <p>This method uses a backward loop. It is significantly faster than the corresponding
  * method in {@link java.util.Arrays}.
  *
  * @param a1 a big array.
  * @param a2 another big array.
  * @return true if the two big arrays are of the same length, and their elements are equal.
  */

 public static KEY_GENERIC boolean equals(final KEY_GENERIC_TYPE[][] a1, final KEY_GENERIC_TYPE
a2[][]) {
  if (length(a1) != length(a2)) return false;
  int i = a1.length, j;
  KEY_GENERIC_TYPE[] t, u;
  while(i-- != 0) {
   t = a1[i];
   u = a2[i];
   j = t.length;
   while(j-- != 0) if (! KEY_EQUALS(t[j], u[j])) return false;
  }
  return true;
 }

 /* Returns a string representation of the contents of the specified big array.
  *
  * The string representation consists of a list of the big array's elements, enclosed in square brackets ("[]"). Adjacent
elements are separated by the characters ", " (a comma followed by a space). Returns "null" if {@code a} is null.
  * @param a the big
 array whose string representation to return.
  * @return the string representation of {@code a}.
  */

 public static KEY_GENERIC String toString(final KEY_GENERIC_TYPE[][] a) {
  if (a == null) return "null";
  final long last = length(a) - 1;
  if (last == - 1) return "[]";
  final StringBuilder b = new StringBuilder();
  b.append('[');
  for (long i = 0; ; i++) {
   b.append(String.valueOf(get(a, i)));
   if (i == last) return b.append(']').toString();
   b.append(", ");
  }
```

```
 }


 /** Ensures that a range given by its first (inclusive) and last (exclusive) elements fits a big array.
  *
  * <p>This method may be used whenever a big array range check is needed.
  *
  * @param a a big array.
  * @param from a start index (inclusive).
  * @param to an end index (inclusive).
  * @throws IllegalArgumentException if {@code from} is greater than {@code to}.
  * @throws ArrayIndexOutOfBoundsException if {@code from} or {@code to} are greater than the big array
length or negative.
  */
 public
 static KEY_GENERIC void ensureFromTo(final KEY_GENERIC_TYPE[][] a, final long from, final long to) {
  BigArrays.ensureFromTo(length(a), from, to);
 }

 /** Ensures that a range given by an offset and a length fits a big array.
  *
  * <p>This method may be used whenever a big array range check is needed.
  *
  * @param a a big array.
  * @param offset a start index.
  * @param length a length (the number of elements in the range).
  * @throws IllegalArgumentException if {@code length} is negative.
  * @throws ArrayIndexOutOfBoundsException if {@code offset} is negative or {@code offset}+{@code length} is
greater than the big array length.
  */
 public static KEY_GENERIC void ensureOffsetLength(final KEY_GENERIC_TYPE[][] a, final long offset, final
long length) {
  BigArrays.ensureOffsetLength(length(a), offset, length);
 }

 /** Ensures that two big arrays are of the same length.
  *
  * @param a a big array.
  * @param b another big array.
  * @throws IllegalArgumentException if
 the two argument arrays are not of the same length.
  */
 public static KEY_GENERIC void ensureSameLength(final KEY_GENERIC_TYPE[][] a, final
KEY_GENERIC_TYPE[][] b) {
  if (length(a) != length(b)) throw new IllegalArgumentException("Array size mismatch: " + length(a) + " != " +
length(b));
 }
```

```
 /** Shuffles the specified big array fragment using the specified pseudorandom number generator.
  *
  * @param a the big array to be shuffled.
  * @param from the index of the first element (inclusive) to be shuffled.
  * @param to the index of the last element (exclusive) to be shuffled.
  * @param random a pseudorandom number generator.
  * @return {@code a}.
  */
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] shuffle(final KEY_GENERIC_TYPE[][] a, final long
from, final long to, final Random random) {
  for(long i = to - from; i-- != 0;) {
   final long p = (random.nextLong() & 0x7FFFFFFFFFFFFFFL) % (i + 1);
   final KEY_GENERIC_TYPE t = BigArrays.get(a, from + i);
   BigArrays.set(a, from
 + i, BigArrays.get(a, from + p));
   BigArrays.set(a, from + p, t);
  }
  return a;
 }

 /** Shuffles the specified big array using the specified pseudorandom number generator.
  *
  * @param a the big array to be shuffled.
  * @param random a pseudorandom number generator.
  * @return {@code a}.
  */
 public static KEY_GENERIC KEY_GENERIC_TYPE[][] shuffle(final KEY_GENERIC_TYPE[][] a, final Random
random) {
  for(long i = length(a); i-- != 0;) {
   final long p = (random.nextLong() & 0x7FFFFFFFFFFFFFFL) % (i + 1);
   final KEY_GENERIC_TYPE t = BigArrays.get(a, i);
   BigArrays.set(a, i, BigArrays.get(a, p));
   BigArrays.set(a, p, t);
  }
  return a;
 }
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/BigArraysFragment.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
```

```
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/



#if KEY_CLASS_Byte

// HORRIBLE kluges to work around bug #6478546

private static final int MAX_IO_LENGTH = 1024 * 1024;

private static int read(final InputStream is, final byte a[], final int offset, final int length) throws IOException {
 if (length == 0) return 0;

 int read = 0, result;
 do {
  result = is.read(a,
 offset + read, Math.min(length - read, MAX_IO_LENGTH));
  if (result < 0) return read;
  read += result;
 } while(read < length);

 return read;
}

private static void write(final OutputStream outputStream, final byte a[], final int offset, final int length) throws
IOException {
 int written = 0;
 while(written < length) {
  outputStream.write(a, offset + written, Math.min(length - written, MAX_IO_LENGTH));
  written += Math.min(length - written, MAX_IO_LENGTH);
 }
}

private static void write(final DataOutput dataOutput, final byte a[], final int offset, final int length) throws
IOException {
 int written = 0;
 while(written < length) {
  dataOutput.write(a, offset + written, Math.min(length - written, MAX_IO_LENGTH));
  written += Math.min(length - written, MAX_IO_LENGTH);
 }
```

```
}

// Additional read/write methods to work around the DataInput/DataOutput schizophrenia.

/** Loads bytes from a given input stream, storing them in a given array fragment.
*
* <p>Note that this method is
 going to be significantly faster than {@link #loadBytes(DataInput,byte[],int,int)}
* as it uses {@link InputStream}'s bulk-read methods.
*
* @param inputStream an input stream.
* @param array an array which will be filled with data from {@code inputStream}.
* @param offset the index of the first element of {@code array} to be filled.
* @param length the number of elements of {@code array} to be filled.
* @return the number of elements actually read from {@code inputStream} (it might be less than {@code length} if
{@code inputStream} ends).
*/
public static int LOAD_KEYS(final InputStream inputStream, final KEY_TYPE[] array, final int offset, final int
length) throws IOException {
 return read(inputStream, array, offset, length);
}

/** Loads bytes from a given input stream, storing them in a given array.
*
* <p>Note that this method is going to be significantly faster than {@link #loadBytes(DataInput,byte[])}
* as it uses {@link InputStream}'s bulk-read methods.
*
* @param
 inputStream an input stream.
* @param array an array which will be filled with data from {@code inputStream}.
* @return the number of elements actually read from {@code inputStream} (it might be less than the array length if
{@code inputStream} ends).
*/
public static int LOAD_KEYS(final InputStream inputStream, final KEY_TYPE[] array) throws IOException {
 return read(inputStream, array, 0, array.length);
}

/** Stores an array fragment to a given output stream.
*
* <p>Note that this method is going to be significantly faster than {@link #storeBytes(byte[],int,int,DataOutput)}
* as it uses {@link OutputStream}'s bulk-read methods.
*
* @param array an array whose elements will be written to {@code outputStream}.
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param outputStream an output stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final int offset,
```

```
 final int length, final OutputStream outputStream) throws IOException {
 write(outputStream, array, offset, length);
}

/** Stores an array to a given output stream.
*
* <p>Note that this method is going to be significantly faster than {@link #storeBytes(byte[],DataOutput)}
* as it uses {@link OutputStream}'s bulk-read methods.
*
* @param array an array whose elements will be written to {@code outputStream}.
* @param outputStream an output stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final OutputStream outputStream) throws IOException
{
 write(outputStream, array, 0, array.length);
}


private static long read(final InputStream is, final byte a[][], final long offset, final long length) throws IOException
{
 if (length == 0) return 0;

 long read = 0;
 int segment = segment(offset);
 int displacement = displacement(offset);
 int result;
 do {
  result = is.read(a[segment], displacement, (int)Math.min(a[segment].length - displacement, Math.min(length - read,
 MAX_IO_LENGTH)));
  if (result < 0) return read;
  read += result;
  displacement += result;
  if (displacement == a[segment].length) {
   segment++;
   displacement = 0;
  }
 } while(read < length);

 return read;
}

private static void write(final OutputStream outputStream, final byte a[][], final long offset, final long length) throws
IOException {
 if (length == 0) return;
 long written = 0;
 int toWrite;
 int segment = segment(offset);
 int displacement = displacement(offset);
```

```
 do {
  toWrite = (int)Math.min(a[segment].length - displacement, Math.min(length - written, MAX_IO_LENGTH));
  outputStream.write(a[segment], displacement, toWrite);
  written += toWrite;
  displacement += toWrite;
  if (displacement == a[segment].length) {
   segment++;
   displacement = 0;
  }
 } while(written < length);
}

private static void write(final DataOutput dataOutput, final byte a[][], final long offset, final long length) throws
IOException {
 if (length == 0) return;
 long written = 0;
 int toWrite;
 int
 segment = segment(offset);
 int displacement = displacement(offset);
 do {
  toWrite = (int)Math.min(a[segment].length - displacement, Math.min(length - written, MAX_IO_LENGTH));
  dataOutput.write(a[segment], displacement, toWrite);
  written += toWrite;
  displacement += toWrite;
  if (displacement == a[segment].length) {
   segment++;
   displacement = 0;
  }
 } while(written < length);
}

// Additional read/write methods to work around the DataInput/DataOutput schizophrenia.


/** Loads bytes from a given input stream, storing them in a given big-array fragment.
*
* <p>Note that this method is going to be significantly faster than {@link
#loadBytes(DataInput,byte[][],long,long)}
* as it uses {@link InputStream}'s bulk-read methods.
*
* @param inputStream an input stream.
* @param array a big array which will be filled with data from {@code inputStream}.
* @param offset the index of the first element of {@code array} to be filled.
* @param length the number of elements
 of {@code array} to be filled.
* @return the number of elements actually read from {@code inputStream} (it might be less than {@code length} if
{@code inputStream} ends).
```

```
*/
public static long LOAD_KEYS(final InputStream inputStream, final KEY_TYPE[][] array, final long offset, final long length) throws IOException {
 return read(inputStream, array, offset, length);
}

/** Loads bytes from a given input stream, storing them in a given big array.
*
* <p>Note that this method is going to be significantly faster than {@link #loadBytes(DataInput,byte[][])}
* as it uses {@link InputStream}'s bulk-read methods.
*
* @param inputStream an input stream.
* @param array a big array which will be filled with data from {@code inputStream}.
* @return the number of elements actually read from {@code inputStream} (it might be less than the array length if {@code inputStream} ends).
*/
public static long LOAD_KEYS(final InputStream inputStream, final KEY_TYPE[][] array) throws IOException
 {
 return read(inputStream, array, 0, length(array));
}

/** Stores a big-array fragment to a given output stream.
*
* <p>Note that this method is going to be significantly faster than {@link #storeBytes(byte[][],long,long,DataOutput)}
* as it uses {@link OutputStream}'s bulk-read methods.
*
* @param array a big array whose elements will be written to {@code outputStream}.
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param outputStream an output stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final long offset, final long length, final OutputStream outputStream) throws IOException {
 write(outputStream, array, offset, length);
}

/** Stores a big array to a given output stream.
*
* <p>Note that this method is going to be significantly faster than {@link #storeBytes(byte[][],DataOutput)}
* as it uses {@link OutputStream}'s bulk-read methods.
*
 * @param array a big array whose elements will be written to {@code outputStream}.
* @param outputStream an output stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final OutputStream outputStream) throws IOException {
 write(outputStream, array, 0, length(array));
```

```
}

#endif


/** Loads elements from a given data input, storing them in a given array fragment.
*
* @param dataInput a data input.
* @param array an array which will be filled with data from {@code dataInput}.
* @param offset the index of the first element of {@code array} to be filled.
* @param length the number of elements of {@code array} to be filled.
* @return the number of elements actually read from {@code dataInput} (it might be less than {@code length} if
{@code dataInput} ends).
*/
public static int LOAD_KEYS(final DataInput dataInput, final KEY_TYPE[] array, final int offset, final int length)
throws IOException {
 PACKAGE.ARRAYS.ensureOffsetLength(array, offset, length);
 int i = 0;
 try
 {
  for(i = 0; i < length; i++) array[i + offset] = dataInput.READ_KEY();
 }
 catch(EOFException itsOk) {}
 return i;
}

/** Loads elements from a given data input, storing them in a given array.
*
* @param dataInput a data input.
* @param array an array which will be filled with data from {@code dataInput}.
* @return the number of elements actually read from {@code dataInput} (it might be less than the array length if
{@code dataInput} ends).
*/
public static int LOAD_KEYS(final DataInput dataInput, final KEY_TYPE[] array) throws IOException {
 int i = 0;
 try {
  final int length = array.length;
  for(i = 0; i < length; i++) array[i] = dataInput.READ_KEY();
 }
 catch(EOFException itsOk) {}
 return i;
}

/** Loads elements from a file given by a {@link File} object, storing them in a given array fragment.
*
* @param file a file.
* @param array an array which will be filled with data from the specified file.
* @param offset the index of the first element of {@code array} to
```

```
 be filled.
* @param length the number of elements of {@code array} to be filled.
* @return the number of elements actually read from the given file (it might be less than {@code length} if the file
is too short).
*/
public static int LOAD_KEYS(final File file, final KEY_TYPE[] array, final int offset, final int length) throws
IOException {
 PACKAGE.ARRAYS.ensureOffsetLength(array, offset, length);

 final FileInputStream fis = new FileInputStream(file);
#if KEY_CLASS_Byte
 final int result = read(fis, array, offset, length);
 fis.close();
 return result;
#else
 final DataInputStream dis = new DataInputStream(new FastBufferedInputStream(fis));

 int i = 0;
 try {
  for(i = 0; i < length; i++) array[i + offset] = dis.READ_KEY();
 }
 catch(EOFException itsOk) {}

 dis.close();
 return i;
#endif
}

/** Loads elements from a file given by a pathname, storing them in a given array fragment.
*
* @param filename a filename.
* @param array an array which will be filled with data from the
 specified file.
* @param offset the index of the first element of {@code array} to be filled.
* @param length the number of elements of {@code array} to be filled.
* @return the number of elements actually read from the given file (it might be less than {@code length} if the file
is too short).
*/
public static int LOAD_KEYS(final CharSequence filename, final KEY_TYPE[] array, final int offset, final int
length) throws IOException {
 return LOAD_KEYS(new File(filename.toString()), array, offset, length);
}

/** Loads elements from a file given by a {@link File} object, storing them in a given array.
*
* @param file a file.
* @param array an array which will be filled with data from the specified file.
* @return the number of elements actually read from the given file (it might be less than the array length if the file
```

```
is too short).
*/
public static int LOAD_KEYS(final File file, final KEY_TYPE[] array) throws IOException {
 final FileInputStream fis = new FileInputStream(file);
#if
 KEY_CLASS_Byte
 final int result = read(fis, array, 0, array.length);
 fis.close();
 return result;
#else
 final DataInputStream dis = new DataInputStream(new FastBufferedInputStream(fis));

 int i = 0;
 try {
  final int length = array.length;
  for(i = 0; i < length; i++) array[i] = dis.READ_KEY();
 }
 catch(EOFException itsOk) {}

 dis.close();

 return i;
#endif
}

/** Loads elements from a file given by a pathname, storing them in a given array.
*
* @param filename a filename.
* @param array an array which will be filled with data from the specified file.
* @return the number of elements actually read from the given file (it might be less than the array length if the file
is too short).
*/
public static int LOAD_KEYS(final CharSequence filename, final KEY_TYPE[] array) throws IOException {
 return LOAD_KEYS(new File(filename.toString()), array);
}

/** Loads elements from a file given by a {@link File} object, storing them in a new array.
*
* <p>Note that the length
 of the returned array will be computed
* dividing the specified file size by the number of bytes used to
* represent each element.
*
* @param file a file.
* @return an array filled with the content of the specified file.
*/
public static KEY_TYPE[] LOAD_KEYS(final File file) throws IOException {
 final FileInputStream fis = new FileInputStream(file);
```

```
#if KEY_CLASS_Boolean
 final long length = fis.getChannel().size();
#else
 final long length = fis.getChannel().size() / (KEY_CLASS.SIZE / 8);
#endif

 if (length > Integer.MAX_VALUE) {
  fis.close();
  throw new IllegalArgumentException("File too long: " + fis.getChannel().size()+ " bytes (" + length + " elements)");
 }

 final KEY_TYPE[] array = new KEY_TYPE[(int)length];

#if KEY_CLASS_Byte
 if (read(fis, array, 0, (int)length) < length) throw new EOFException();
 fis.close();
#else
 final DataInputStream dis = new DataInputStream(new FastBufferedInputStream(fis));
 for(int i = 0; i < length; i++) array[i] = dis.READ_KEY();
 dis.close();
#endif
 return
 array;
}

/** Loads elements from a file given by a filename, storing them in a new array.
*
* <p>Note that the length of the returned array will be computed
* dividing the specified file size by the number of bytes used to
* represent each element.
*
* @param filename a filename.
* @return an array filled with the content of the specified file.
*/
public static KEY_TYPE[] LOAD_KEYS(final CharSequence filename) throws IOException {
 return LOAD_KEYS(new File(filename.toString()));
}

/** Stores an array fragment to a given data output.
*
* @param array an array whose elements will be written to {@code dataOutput}.
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param dataOutput a data output.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final int offset, final int length, final DataOutput
dataOutput) throws IOException {
```

```
 PACKAGE.ARRAYS.ensureOffsetLength(array,
 offset, length);
#if KEY_CLASS_Byte
 write(dataOutput, array, offset, length);
#else
 for(int i = 0; i < length; i++) dataOutput.WRITE_KEY(array[offset + i]);
#endif
}

/** Stores an array to a given data output.
*
* @param array an array whose elements will be written to {@code dataOutput}.
* @param dataOutput a data output.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final DataOutput dataOutput) throws IOException {
#if KEY_CLASS_Byte
 write(dataOutput, array, 0, array.length);
#else
 final int length = array.length;
 for(int i = 0; i < length; i++) dataOutput.WRITE_KEY(array[i]);
#endif
}

/** Stores an array fragment to a file given by a {@link File} object.
*
* @param array an array whose elements will be written to {@code filename}.
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param file a file.
*/
public static void STORE_KEYS(final
 KEY_TYPE array[], final int offset, final int length, final File file) throws IOException {
 PACKAGE.ARRAYS.ensureOffsetLength(array, offset, length);
#if KEY_CLASS_Byte
 final OutputStream os = new FastBufferedOutputStream(new FileOutputStream(file));
 write(os, array, offset, length);
 os.close();
#else
 final DataOutputStream dos = new DataOutputStream(new FastBufferedOutputStream(new
FileOutputStream(file)));
 for(int i = 0; i < length; i++) dos.WRITE_KEY(array[offset + i]);
 dos.close();
#endif
}

/** Stores an array fragment to a file given by a pathname.
*
* @param array an array whose elements will be written to {@code filename}.
```

```
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param filename a filename.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final int offset, final int length, final CharSequence
filename) throws IOException {
 STORE_KEYS(array,
 offset, length, new File(filename.toString()));
}

/** Stores an array to a file given by a {@link File} object.
*
* @param array an array whose elements will be written to {@code filename}.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final File file) throws IOException {
#if KEY_CLASS_Byte
 final OutputStream os = new FastBufferedOutputStream(new FileOutputStream(file));
 write(os, array, 0, array.length);
 os.close();
#else
 final int length = array.length;
 final DataOutputStream dos = new DataOutputStream(new FastBufferedOutputStream(new
FileOutputStream(file)));
 for(int i = 0; i < length; i++) dos.WRITE_KEY(array[i]);
 dos.close();
#endif
}

/** Stores an array to a file given by a pathname.
*
* @param array an array whose elements will be written to {@code filename}.
* @param filename a filename.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final CharSequence filename) throws IOException {
 STORE_KEYS(array, new File(filename.toString()));
}




/**
 Loads elements from a given data input, storing them in a given big-array fragment.
*
* @param dataInput a data input.
* @param array a big array which will be filled with data from {@code dataInput}.
* @param offset the index of the first element of {@code bigArray} to be filled.
```

```
* @param length the number of elements of {@code bigArray} to be filled.
* @return the number of elements actually read from {@code dataInput} (it might be less than {@code length} if
{@code dataInput} ends).
*/
public static long LOAD_KEYS(final DataInput dataInput, final KEY_TYPE[][] array, final long offset, final long
length) throws IOException {
 ensureOffsetLength(array, offset, length);
 long c = 0;
 try {
  for(int i = segment(offset); i < segment(offset + length + SEGMENT_MASK); i++) {
   final KEY_TYPE[] t = array[i];
   final int l = (int)Math.min(t.length, offset + length - start(i));
   for(int d = (int)Math.max(0, offset - start(i)); d < l; d++) {
    t[d]
 = dataInput.READ_KEY();
    c++;
   }
  }
 }
 catch(EOFException itsOk) {}
 return c;
}

/** Loads elements from a given data input, storing them in a given big array.
*
* @param dataInput a data input.
* @param array a big array which will be filled with data from {@code dataInput}.
* @return the number of elements actually read from {@code dataInput} (it might be less than the array length if
{@code dataInput} ends).
*/
public static long LOAD_KEYS(final DataInput dataInput, final KEY_TYPE[][] array) throws IOException {
 long c = 0;
 try {
  for(int i = 0; i < array.length; i++) {
   final KEY_TYPE[] t = array[i];
   final int l = t.length;
   for(int d = 0; d < l; d++) {
    t[d] = dataInput.READ_KEY();
    c++;
   }
  }
 }
 catch(EOFException itsOk) {}
 return c;
}

/** Loads elements from a file given by a {@link File} object, storing them in a given big-array fragment.
*
```

```
* @param file a file.
* @param array a big array which will be filled with data from the specified
 file.
* @param offset the index of the first element of {@code array} to be filled.
* @param length the number of elements of {@code array} to be filled.
* @return the number of elements actually read from the given file (it might be less than {@code length} if the file
is too short).
*/
public static long LOAD_KEYS(final File file, final KEY_TYPE[][] array, final long offset, final long length)
throws IOException {
 ensureOffsetLength(array, offset, length);

 final FileInputStream fis = new FileInputStream(file);
#if KEY_CLASS_Byte
 final long result = read(fis, array, offset, length);
 fis.close();
 return result;
#else
 final DataInputStream dis = new DataInputStream(new FastBufferedInputStream(fis));

 long c = 0;
 try {
  for(int i = segment(offset); i < segment(offset + length + SEGMENT_MASK); i++) {
   final KEY_TYPE[] t = array[i];
   final int l = (int)Math.min(t.length, offset + length - start(i));
   for(int d = (int)Math.max(0, offset - start(i)); d < l; d++) {
    t[d]
 = dis.READ_KEY();
    c++;
   }
  }
 }
 catch(EOFException itsOk) {}
 dis.close();
 return c;
#endif
}

/** Loads elements from a file given by a pathname, storing them in a given big-array fragment.
*
* @param filename a filename.
* @param array an array which will be filled with data from the specified file.
* @param offset the index of the first element of {@code array} to be filled.
* @param length the number of elements of {@code array} to be filled.
* @return the number of elements actually read from the given file (it might be less than {@code length} if the file
is too short).
*/
public static long LOAD_KEYS(final CharSequence filename, final KEY_TYPE[][] array, final long offset, final
```

```
long length) throws IOException {
 return LOAD_KEYS(new File(filename.toString()), array, offset, length);
}

/** Loads elements from a file given by a {@link File} object, storing them in a given big array.
*
* @param file a file.
* @param array a big array which will be filled with
 data from the specified file.
* @return the number of elements actually read from the given file (it might be less than the array length if the file
is too short).
*/
public static long LOAD_KEYS(final File file, final KEY_TYPE[][] array) throws IOException {
 final FileInputStream fis = new FileInputStream(file);
#if KEY_CLASS_Byte
 final long result = read(fis, array, 0, length(array));
 fis.close();
 return result;
#else
 final DataInputStream dis = new DataInputStream(new FastBufferedInputStream(fis));

 long c = 0;
 try {
  for(int i = 0; i < array.length; i++) {
   final KEY_TYPE[] t = array[i];
   final int l = t.length;
   for(int d = 0; d < l; d++) {
    t[d] = dis.READ_KEY();
    c++;
   }
  }
 }
 catch(EOFException itsOk) {}

 dis.close();

 return c;
#endif
}

/** Loads elements from a file given by a pathname, storing them in a given big array.
*
* @param filename a filename.
* @param array a big array which will be filled with data from the specified file.
* @return
 the number of elements actually read from the given file (it might be less than the array length if the file is too
short).
*/
```

```
public static long LOAD_KEYS(final CharSequence filename, final KEY_TYPE[][] array) throws IOException {
 return LOAD_KEYS(new File(filename.toString()), array);
}

/** Loads elements from a file given by a {@link File} object, storing them in a new big array.
*
* <p>Note that the length of the returned big array will be computed
* dividing the specified file size by the number of bytes used to
* represent each element.
*
* @param file a file.
* @return a big array filled with the content of the specified file.
*/
public static KEY_TYPE[][] LOAD_KEYS_BIG(final File file) throws IOException {
 final FileInputStream fis = new FileInputStream(file);

#if KEY_CLASS_Boolean
 final long length = fis.getChannel().size();
#else
 final long length = fis.getChannel().size() / (KEY_CLASS.SIZE / 8);
#endif

 final KEY_TYPE[][] array = BIG_ARRAYS.newBigArray(length);

#if
 KEY_CLASS_Byte
 if (read(fis, array, 0, length) < length) throw new EOFException();
 fis.close();
#else
 final DataInputStream dis = new DataInputStream(new FastBufferedInputStream(fis));

 for(int i = 0; i < array.length; i++) {
  final KEY_TYPE[] t = array[i];
  final int l = t.length;
  for(int d = 0; d < l; d++) t[d] = dis.READ_KEY();
 }

 dis.close();
#endif
 return array;
}

/** Loads elements from a file given by a filename, storing them in a new big array.
*
* <p>Note that the length of the returned big array will be computed
* dividing the specified file size by the number of bytes used to
* represent each element.
*
```

```
* @param filename a filename.
* @return a big array filled with the content of the specified file.
*/
public static KEY_TYPE[][] LOAD_KEYS_BIG(final CharSequence filename) throws IOException {
 return LOAD_KEYS_BIG(new File(filename.toString()));
}

/** Stores an array fragment to a given data output.
*
* @param array an array whose elements will
 be written to {@code dataOutput}.
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param dataOutput a data output.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final long offset, final long length, final DataOutput
dataOutput) throws IOException {
 ensureOffsetLength(array, offset, length);
#if KEY_CLASS_Byte
 write(dataOutput, array, offset, length);
#else
 for(int i = segment(offset); i < segment(offset + length + SEGMENT_MASK); i++) {
  final KEY_TYPE[] t = array[i];
  final int l = (int)Math.min(t.length, offset + length - start(i));
  for(int d = (int)Math.max(0, offset - start(i)); d < l; d++) dataOutput.WRITE_KEY(t[d]);
 }
#endif
}

/** Stores a big array to a given data output.
*
* @param array a big array whose elements will be written to {@code dataOutput}.
* @param dataOutput a data output.
*/
public static void STORE_KEYS(final KEY_TYPE array[][],
 final DataOutput dataOutput) throws IOException {
#if KEY_CLASS_Byte
 write(dataOutput, array, 0, length(array));
#else
 for(int i = 0; i < array.length; i++) {
  final KEY_TYPE[] t = array[i];
  final int l = t.length;
  for(int d = 0; d < l; d++) dataOutput.WRITE_KEY(t[d]);
 }
#endif
}

/** Stores a big-array fragment to a file given by a {@link File} object.
```

```
*
* @param array a big array whose elements will be written to {@code filename}.
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final long offset, final long length, final File file)
throws IOException {
 ensureOffsetLength(array, offset, length);
#if KEY_CLASS_Byte
 final OutputStream os = new FastBufferedOutputStream(new FileOutputStream(file));
 write(os, array, offset, length);
 os.close();
#else
 final DataOutputStream dos = new DataOutputStream(new
 FastBufferedOutputStream(new FileOutputStream(file)));
 for(int i = segment(offset); i < segment(offset + length + SEGMENT_MASK); i++) {
  final KEY_TYPE[] t = array[i];
  final int l = (int)Math.min(t.length, offset + length - start(i));
  for(int d = (int)Math.max(0, offset - start(i)); d < l; d++) dos.WRITE_KEY(t[d]);
 }
 dos.close();

#endif
}

/** Stores a big-array fragment to a file given by a pathname.
*
* @param array a big array whose elements will be written to {@code filename}.
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param filename a filename.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final long offset, final long length, final
CharSequence filename) throws IOException {
 STORE_KEYS(array, offset, length, new File(filename.toString()));
}

/** Stores an array to a file given by a {@link File} object.
*
* @param array an array
 whose elements will be written to {@code filename}.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final File file) throws IOException {
#if KEY_CLASS_Byte
 final OutputStream os = new FastBufferedOutputStream(new FileOutputStream(file));
 write(os, array, 0, length(array));
```

```
 os.close();
#else
 final DataOutputStream dos = new DataOutputStream(new FastBufferedOutputStream(new
FileOutputStream(file)));
 for(int i = 0; i < array.length; i++) {
  final KEY_TYPE[] t = array[i];
  final int l = t.length;
  for(int d = 0; d < l; d++) dos.WRITE_KEY(t[d]);
 }
 dos.close();
#endif
}

/** Stores a big array to a file given by a pathname.
*
* @param array a big array whose elements will be written to {@code filename}.
* @param filename a filename.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final CharSequence filename) throws IOException {
 STORE_KEYS(array, new File(filename.toString()));
}

/** Stores the element returned by an iterator to
 a given data output.
*
* @param i an iterator whose output will be written to {@code dataOutput}.
* @param dataOutput a filename.
*/
public static void STORE_KEYS(final KEY_ITERATOR i, final DataOutput dataOutput) throws IOException {
 while(i.hasNext()) dataOutput.WRITE_KEY(i.NEXT_KEY());
}

/** Stores the element returned by an iterator to a file given by a {@link File} object.
*
* @param i an iterator whose output will be written to {@code filename}.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_ITERATOR i, final File file) throws IOException {
 final DataOutputStream dos = new DataOutputStream(new FastBufferedOutputStream(new
FileOutputStream(file)));
 while(i.hasNext()) dos.WRITE_KEY(i.NEXT_KEY());
 dos.close();
}

/** Stores the element returned by an iterator to a file given by a pathname.
*
* @param i an iterator whose output will be written to {@code filename}.
* @param filename a filename.
```

```
*/
public static void STORE_KEYS(final KEY_ITERATOR i,
 final CharSequence filename) throws IOException {
 STORE_KEYS(i, new File(filename.toString()));
}

/** A wrapper that exhibits the content of a data input stream as a type-specific iterator. */

private static final class KEY_DATA_INPUT_WRAPPER implements KEY_ITERATOR {
 private final DataInput dataInput;
 private boolean toAdvance = true;
 private boolean endOfProcess = false;
 private KEY_TYPE next;

 public KEY_DATA_INPUT_WRAPPER(final DataInput dataInput) {
  this.dataInput = dataInput;
 }

 @Override
 public boolean hasNext() {
  if (! toAdvance) return ! endOfProcess;

  toAdvance = false;

  try { next = dataInput.READ_KEY(); }
  catch(EOFException eof) { endOfProcess = true; }
  catch(IOException rethrow) { throw new RuntimeException(rethrow); }

  return ! endOfProcess;
 }

 @Override
 public KEY_TYPE NEXT_KEY() {
  if (! hasNext()) throw new NoSuchElementException();
  toAdvance = true;
  return next;
 }
}

/** Wraps the given data input stream into an iterator.
*
* @param
 dataInput a data input.
*/
public static KEY_ITERATOR AS_KEY_ITERATOR(final DataInput dataInput) {
 return new KEY_DATA_INPUT_WRAPPER(dataInput);
}
```

```
/** Wraps a file given by a {@link File} object into an iterator.
*
* @param file a file.
*/
public static KEY_ITERATOR AS_KEY_ITERATOR(final File file) throws IOException {
 return new KEY_DATA_INPUT_WRAPPER(new DataInputStream(new FastBufferedInputStream(new
FileInputStream(file))));
}

/** Wraps a file given by a pathname into an iterator.
*
* @param filename a filename.
*/
public static KEY_ITERATOR AS_KEY_ITERATOR(final CharSequence filename) throws IOException {
 return AS_KEY_ITERATOR(new File(filename.toString()));
}

/** Wraps a file given by a {@link File} object into an iterable object.
*
* @param file a file.
*/
public static KEY_ITERABLE AS_KEY_ITERABLE(final File file) {
 return () -> {
  try { return AS_KEY_ITERATOR(file); }
  catch(IOException e) { throw new RuntimeException(e); }
 };
}

/** Wraps a file given by
 a pathname into an iterable object.
*
* @param filename a filename.
*/
public static KEY_ITERABLE AS_KEY_ITERABLE(final CharSequence filename) {
 return () -> {
  try { return AS_KEY_ITERATOR(filename); }
  catch(IOException e) { throw new RuntimeException(e); }
 };
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/BinIOFragment.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2017 Sebastiano Vigna
*
```

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericLinkedOpenHashTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntLinkedOpenHashSetTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericLinkedOpenHashTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntOpenCustomHashSetTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/ObjectOpenCustomHashSetTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/IntLinkedOpenCustomHashSetTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2IntLinkedOpenHashMapTest.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/ints/Int2IntOpenCustomHashMapTest.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

```
*/


package PACKAGE;

import java.util.Collection;
import java.util.Set;

/** A class providing static methods and objects that do useful things with type-specific sets.
*
* @see java.util.Collections
*/

public final class SETS {

 private SETS() {}

 /** An immutable class representing the empty set and implementing
 a type-specific set interface.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific set.
  */

 public static class EmptySet KEY_GENERIC extends COLLECTIONS.EmptyCollection KEY_GENERIC
implements SET KEY_GENERIC, java.io.Serializable, Cloneable {
  private static final long serialVersionUID = -7046029254386353129L;

  protected EmptySet() {}

  @Override
  public boolean remove(KEY_TYPE ok) { throw new UnsupportedOperationException(); }

  @Override
  public Object clone() { return EMPTY_SET; }

  @Override
  @SuppressWarnings("rawtypes")
  public boolean equals(final Object o) { return o instanceof Set && ((Set)o).isEmpty(); }

#if KEYS_PRIMITIVE
  @Deprecated
  @Override
  public boolean rem(final KEY_TYPE k) { return super.rem(k); }
#endif

  private Object readResolve() { return EMPTY_SET; }
 }
```

```
 /** An empty set (immutable). It is serializable and cloneable.
  */
 SUPPRESS_WARNINGS_KEY_RAWTYPES
 public static final EmptySet EMPTY_SET
 = new EmptySet();

#if KEYS_REFERENCE
 /** Returns an empty set (immutable). It is serializable and cloneable.
  *
  * <p>This method provides a typesafe access to {@link #EMPTY_SET}.
  * @return an empty set (immutable).
  */
 @SuppressWarnings("unchecked")
 public static KEY_GENERIC SET KEY_GENERIC emptySet() {
  return EMPTY_SET;
 }
#endif


 /** An immutable class representing a type-specific singleton set.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific set.
  */

 public static class Singleton KEY_GENERIC extends ABSTRACT_SET KEY_GENERIC implements
java.io.Serializable, Cloneable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final KEY_GENERIC_TYPE element;

  protected Singleton(final KEY_GENERIC_TYPE element) {
   this.element = element;
  }

  @Override
  public boolean contains(final KEY_TYPE k) { return KEY_EQUALS(k, element); }

  @Override
  public boolean remove(final KEY_TYPE k) { throw
 new UnsupportedOperationException(); }

  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC iterator() { return ITERATORS.singleton(element); }
```

```
  @Override
  public int size() { return 1; }

  @Override
  public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

  @Override
  public boolean removeAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean retainAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE

  @Override
  public boolean addAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean removeAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean retainAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

#endif
  @Override
  public Object clone() { return this; }
 }

 /** Returns a type-specific immutable set containing
 only the specified element. The returned set is serializable and cloneable.
  *
  * @param element the only element of the returned set.
  * @return a type-specific immutable set containing just {@code element}.
  */

 public static KEY_GENERIC SET KEY_GENERIC singleton(final KEY_GENERIC_TYPE element) {
  return new Singleton KEY_GENERIC_DIAMOND(element);
 }

#if KEYS_PRIMITIVE

 /** Returns a type-specific immutable set containing only the specified element. The returned set is serializable and
cloneable.
  *
  * @param element the only element of the returned set.
  * @return a type-specific immutable set containing just {@code element}.
  */
```

```
 public static KEY_GENERIC SET KEY_GENERIC singleton(final KEY_GENERIC_CLASS element) {
  return new Singleton KEY_GENERIC_DIAMOND(KEY_CLASS2TYPE(element));
 }

#endif

 /** A synchronized wrapper class for sets. */

 public static class SynchronizedSet KEY_GENERIC extends COLLECTIONS.SynchronizedCollection
KEY_GENERIC implements SET KEY_GENERIC,
 java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected SynchronizedSet(final SET KEY_GENERIC s, final Object sync) {
   super(s, sync);
  }

  protected SynchronizedSet(final SET KEY_GENERIC s) {
   super(s);
  }

  @Override
  public boolean remove(final KEY_TYPE k) { synchronized(sync) { return collection.REMOVE(k); } }

#if KEYS_PRIMITIVE
  @Deprecated
  @Override
  public boolean rem(final KEY_TYPE k) { return super.rem(k); }
#endif
 }


 /** Returns a synchronized type-specific set backed by the given type-specific set.
  *
  * @param s the set to be wrapped in a synchronized set.
  * @return a synchronized view of the specified set.
  * @see java.util.Collections#synchronizedSet(Set)
  */
 public static KEY_GENERIC SET KEY_GENERIC synchronize(final SET KEY_GENERIC s) { return new
SynchronizedSet KEY_GENERIC_DIAMOND(s); }

 /** Returns a synchronized type-specific set backed by the given type-specific set, using an assigned
 object to synchronize.
  *
  * @param s the set to be wrapped in a synchronized set.
  * @param sync an object that will be used to synchronize the access to the set.
```

```
  * @return a synchronized view of the specified set.
  * @see java.util.Collections#synchronizedSet(Set)
  */

 public static KEY_GENERIC SET KEY_GENERIC synchronize(final SET KEY_GENERIC s, final Object sync) {
return new SynchronizedSet KEY_GENERIC_DIAMOND(s, sync); }



 /** An unmodifiable wrapper class for sets. */

 public static class UnmodifiableSet KEY_GENERIC extends COLLECTIONS.UnmodifiableCollection
KEY_GENERIC implements SET KEY_GENERIC, java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected UnmodifiableSet(final SET KEY_GENERIC s) {
   super(s);
  }

  @Override
  public boolean remove(final KEY_TYPE k) { throw new UnsupportedOperationException(); }

  @Override
  public boolean equals(final Object o) { if (o == this) return true; return collection.equals(o);
 }

  @Override
  public int hashCode() { return collection.hashCode(); }

#if KEYS_PRIMITIVE
  @Deprecated
  @Override
  public boolean rem(final KEY_TYPE k) { return super.rem(k); }
#endif
 }


 /** Returns an unmodifiable type-specific set backed by the given type-specific set.
  *
  * @param s the set to be wrapped in an unmodifiable set.
  * @return an unmodifiable view of the specified set.
  * @see java.util.Collections#unmodifiableSet(Set)
  */
 public static KEY_GENERIC SET KEY_GENERIC unmodifiable(final SET KEY_GENERIC s) { return new
UnmodifiableSet KEY_GENERIC_DIAMOND(s); }
```

```
#ifdef TEST

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif KEY_CLASS_Object
  return Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }


 private static void test() {
  int n = 100;
  int c;
  KEY_TYPE k = genKey();
  Singleton
 m = new Singleton(k);
  Set t = java.util.Collections.singleton(KEY2OBJ(k));

  long ms;
  boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement, mThrowsIndex, tThrowsIndex,
mThrowsUnsupp, tThrowsUnsupp;
  boolean rt = false, rm = false;


  /* Now we check that m and t are equal. */
  if (!m.equals(t) || ! t.equals(m)) System.err.println("m: " + m + " t: " + t);

  ensure(m.equals(t), "Error (" + seed + "): ! m.equals(t) at start");
  ensure(t.equals(m), "Error (" + seed + "): ! t.equals(m) at start");

  /* Now we check that m actually holds that data. */
  for(java.util.Iterator i=t.iterator(); i.hasNext();) {
   ensure(m.contains(i.next()), "Error (" + seed + "): m and t differ on an entry after insertion (iterating on t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */
  for(java.util.Iterator i=m.iterator(); i.hasNext();) {
   ensure(t.contains(i.next()), "Error (" + seed + "): m and t differ on an entry after insertion
 (iterating on m)");
  }

  /* Now we check that inquiries about random data give the same answer in m and t. For
```

```
    m we use the polymorphic method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    m.contains(T);
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }

   try {
    t.contains(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + seed + "): contains() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement
 + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + seed + "): contains() divergence in IllegalArgumentException
for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex) ensure(m.contains(KEY2OBJ(T)) ==
t.contains(KEY2OBJ(T)), "Error (" + seed + "): divergence in keys between t and m (polymorphic method) " + m);
  }

  /* Again, we check that inquiries about random data give the same answer in m and t, but
    for m we use the standard method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    m.contains(KEY2OBJ(T));
   }
   catch
 (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
```

```
  catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
  catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

  try {
   t.contains(KEY2OBJ(T));
  }
  catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
  catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + seed + "): contains() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + seed + "): contains() divergence in IllegalArgumentException
for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
  ensure(mThrowsIndex
 == tThrowsIndex, "Error (" + seed + "): contains() divergence in IndexOutOfBoundsException for " + T + " (" +
mThrowsIndex + ", " + tThrowsIndex + ") " + m);
  ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + seed + "): contains() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
  if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp)
ensure(m.contains(KEY2OBJ(T)) ==  t.contains(KEY2OBJ(T)), "Error (" + seed + "): divergence between t and m
(standard method) " + m);
 }

 /* Now we add and remove random data in m and t, checking that the result is the same. */

 for(int i=0; i<20*n;  i++) {
  KEY_TYPE T = genKey();

  mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

  try {
   rm = m.add(KEY2OBJ(T));
  }
  catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException
 e) { mThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
  catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

  try {
   rt = t.add(KEY2OBJ(T));
  }
  catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
```

```
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + seed + "): add() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + seed + "): add() divergence in IllegalArgumentException for "
+ T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + seed + "): add() divergence in IndexOutOfBoundsException
for " +
 T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + seed + "): add() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + seed + "): divergence in add() between t and m " + m);

   T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    rm = m.remove(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    rt = t.remove(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement
 = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   if (! KEY_EQUALS(T, k) && mThrowsUnsupp && ! tThrowsUnsupp) mThrowsUnsupp = false; // Stupid bug in
Collections.singleton()

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + seed + "): remove() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + seed + "): remove() divergence in IllegalArgumentException
for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + seed + "): remove() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + seed + "): remove() divergence in
UnsupportedOperationException
```

```
 for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + seed + "): divergence in remove() between t and m " + m);
  }

  ensure(m.equals(t), "Error (" + seed + "): ! m.equals(t) after removal " + m);
  ensure(t.equals(m), "Error (" + seed + "): ! t.equals(m) after removal " + m);

  /* Now we add and remove random collections in m and t, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    rm = m.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex
 = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    rt = t.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
   }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + seed + "): addAll() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + seed + "): addAll() divergence in IllegalArgumentException
for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + seed + "): addAll() divergence in IndexOutOfBoundsException
for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp
 == tThrowsUnsupp, "Error (" + seed + "): addAll() divergence in UnsupportedOperationException for " + T + " ("
+ mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + seed + "): divergence in addAll() between t and m " + m);

   T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
```

```
mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    rm = m.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    rt = t.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
   }
   catch (java.util.NoSuchElementException e)
 { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   if (! KEY_EQUALS(T, k) && mThrowsUnsupp && ! tThrowsUnsupp) mThrowsUnsupp = false; // Stupid bug in
Collections.singleton()

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + seed + "): removeAll() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + seed + "): removeAll() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + seed + "): removeAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + seed + "): removeAll()
 divergence in UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " +
m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + seed + "): divergence in removeAll() between t and m " + m);
  }

  ensure(m.equals(t), "Error (" + seed + "): ! m.equals(t) after set removal " + m);
  ensure(t.equals(m), "Error (" + seed + "): ! t.equals(m) after set removal " + m);

  /* Now we check that m actually holds the same data. */

  for(java.util.Iterator i=t.iterator(); i.hasNext();) {
   ensure(m.contains(i.next()), "Error (" + seed + "): m and t differ on an entry after removal (iterating on t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */

  for(java.util.Iterator i=m.iterator(); i.hasNext();) {
   ensure(t.contains(i.next()), "Error (" + seed + "): m and t differ on an entry after removal (iterating on m)");
```

```
  }

  if (m instanceof Singleton) {
   ensure(m.equals(((Singleton)m).clone()),
 "Error (" + seed + "): m does not equal m.clone()");
   ensure(((Singleton)m).clone().equals(m), "Error (" + seed + "): m.clone() does not equal m");
  }

  int h = m.hashCode();

  /* Now we save and read m. */

  SET m2 = null;

  try {
   java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
   java.io.OutputStream os = new java.io.FileOutputStream(ff);
   java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

   oos.writeObject(m);
   oos.close();

   java.io.InputStream is = new java.io.FileInputStream(ff);
   java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

   m2 = (SET)ois.readObject();
   ois.close();
   ff.delete();
  }
  catch(Exception e) {
   e.printStackTrace();
   System.exit(1);
  }

#if ! KEY_CLASS_Reference

  ensure(m2.hashCode() == h, "Error (" + seed + "): hashCode() changed after save/read");

  /* Now we check that m2 actually holds that data.
 */

  ensure(m2.equals(t), "Error (" + seed + "): ! m2.equals(t) after save/read");
  ensure(t.equals(m2), "Error (" + seed + "): ! t.equals(m2) after save/read");
#endif

  System.out.println("Test OK");
  return;
 }
```

```
 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);

 private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
 private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

 private static String format(double d) {
  StringBuffer s = new StringBuffer();
  return format.format(d, s, fp).toString();
 }

 private static void fatal(String msg) {
  System.out.println(msg);
  System.exit(1);
 }

 private static void ensure(boolean cond, String msg) {
  if (cond) return;
  fatal(msg);
 }

 /** This method expects as first argument a lower-cased type (e.g., "int"),
  * and as second optional argument a seed. */

 public static void main(String arg[]) throws Exception
 {
  if (arg.length > 1) r = new java.util.Random(seed = Long.parseLong(arg[1]));

  try {
   test();
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed: " + seed);
  }
 }

#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Sets.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2010-2020 Sebastiano Vigna
```

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/BigSwapper.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/BigListIterator.java
*
 /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/Swapper.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/BigList.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/Size64.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package PACKAGE;

import VALUE_PACKAGE.VALUE_COLLECTION;

```
import it.unimi.dsi.fastutil.objects.ObjectSortedSet;
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;

import java.util.Map;
import java.util.SortedMap;

#if KEYS_REFERENCE
import java.util.Comparator;
#endif

/** A type-specific {@link SortedMap};
 provides some additional methods that use polymorphism to avoid (un)boxing.
*
* <p>Additionally, this interface strengthens {@link #entrySet()},
* {@link #keySet()}, {@link #values()},
* {@link #comparator()}, {@link SortedMap#subMap(Object,Object)}, {@link SortedMap#headMap(Object)} and
{@link SortedMap#tailMap(Object)}.
*
* @see SortedMap
*/

public interface SORTED_MAP KEY_VALUE_GENERIC extends MAP KEY_VALUE_GENERIC,
SortedMap<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS> {

 /** Returns a view of the portion of this sorted map whose keys range from {@code fromKey}, inclusive, to
{@code toKey}, exclusive.
  *
  * <p>Note that this specification strengthens the one given in {@link SortedMap#subMap(Object,Object)}.
  *
  * @see SortedMap#subMap(Object,Object)
  */
#if KEYS_REFERENCE
 @Override
#endif
 SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_TYPE fromKey, KEY_GENERIC_TYPE
toKey);

 /** Returns a view of the portion of this sorted map whose keys are strictly less than {@code
 toKey}.
  *
  * <p>Note that this specification strengthens the one given in {@link SortedMap#headMap(Object)}.
  *
  * @see SortedMap#headMap(Object)
  */
#if KEYS_REFERENCE
 @Override
#endif
```

```
 SORTED_MAP KEY_VALUE_GENERIC headMap(KEY_GENERIC_TYPE toKey);

 /** Returns a view of the portion of this sorted map whose keys are greater than or equal to {@code fromKey}.
  *
  * <p>Note that this specification strengthens the one given in {@link SortedMap#tailMap(Object)}.
  *
  * @see SortedMap#tailMap(Object)
  */
#if KEYS_REFERENCE
 @Override
#endif
 SORTED_MAP KEY_VALUE_GENERIC tailMap(KEY_GENERIC_TYPE fromKey);

#if KEYS_PRIMITIVE

 /** Returns the first (lowest) key currently in this map.
  * @see SortedMap#firstKey()
  */
 KEY_GENERIC_TYPE FIRST_KEY();

 /** Returns the last (highest) key currently in this map.
  * @see SortedMap#lastKey()
  */
 KEY_GENERIC_TYPE LAST_KEY();

 /** {@inheritDoc}
  * <p>Note that this specification strengthens the one given in {@link SortedMap#subMap(Object,Object)}.

 * @deprecated Please use the corresponding type-specific method instead.
  */
 @Deprecated
 @Override
 default SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_CLASS from, final
KEY_GENERIC_CLASS to) {
  return subMap(KEY_CLASS2TYPE(from), KEY_CLASS2TYPE(to));
 }

 /** {@inheritDoc}
  * <p>Note that this specification strengthens the one given in {@link SortedMap#headMap(Object)}.
  * @deprecated Please use the corresponding type-specific method instead.
  */
 @Deprecated
 @Override
 default SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_CLASS to) {
  return headMap(KEY_CLASS2TYPE(to));
 }

 /** {@inheritDoc}
```

```
 * <p>Note that this specification strengthens the one given in {@link SortedMap#tailMap(Object)}.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_CLASS from) {
 return tailMap(KEY_CLASS2TYPE(from));
}

/** {@inheritDoc}
 * @deprecated
Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default KEY_GENERIC_CLASS firstKey() {
 return KEY2OBJ(FIRST_KEY());
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default KEY_GENERIC_CLASS lastKey() {
 return KEY2OBJ(LAST_KEY());
}
#endif


/** A sorted entry set providing fast iteration.
 *
 * <p>In some cases (e.g., hash-based classes) iteration over an entry set requires the creation
 * of a large number of entry objects. Some {@code fastutil}
 * maps might return {@linkplain #entrySet() entry set} objects of type {@code FastSortedEntrySet}: in this case,
{@link #fastIterator() fastIterator()}
 * will return an iterator that is guaranteed not to create a large number of objects, <em>possibly
 * by returning always the same entry</em> (of course, mutated).
 */
interface FastSortedEntrySet KEY_VALUE_GENERIC extends ObjectSortedSet<MAP.Entry
KEY_VALUE_GENERIC>, FastEntrySet KEY_VALUE_GENERIC {
 /** {@inheritDoc}
  */
 @Override
 ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> fastIterator();


 /** Returns a fast iterator over this entry set, starting from a given element of the domain (optional operation);
```

```
   * the iterator might return always the same entry instance, suitably mutated.
   *
   * @param from an element to start from.
   * @return a fast iterator over this sorted entry set starting at {@code from}; the iterator might return always the
same entry object, suitably mutated.
   */
  ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> fastIterator(MAP.Entry
KEY_VALUE_GENERIC from);
 }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
 /** Returns a sorted-set view of the mappings contained in this map.
  *  <p>Note that this specification strengthens the one given in the
  *  corresponding type-specific unsorted map.
  *
  * @return a sorted-set view of the mappings contained in this map.
  * @see SortedMap#entrySet()

 * @deprecated Please use the corresponding type-specific method instead.
  */

 @SuppressWarnings({ "unchecked", "rawtypes" })
 @Deprecated
 @Override
 default ObjectSortedSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() {
  return (ObjectSortedSet)ENTRYSET();
 }
#else
 /** Returns a sorted-set view of the mappings contained in this map.
  *  <p>Note that this specification strengthens the one given in the
  *  corresponding type-specific unsorted map.
  *
  * @return a sorted-set view of the mappings contained in this map.
  * @see Map#entrySet()
  */

 @SuppressWarnings({ "unchecked", "rawtypes" })
 @Override
 default ObjectSortedSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() {
  return (ObjectSortedSet)ENTRYSET();
 }
#endif

 /** Returns a type-specific sorted-set view of the mappings contained in this map.
  * <p>Note that this specification strengthens the one given in the
  * corresponding type-specific unsorted map.
  *
  * @return
```

```
a type-specific sorted-set view of the mappings contained in this map.
 * @see #entrySet()
 */

@Override
ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET();

/** Returns a type-specific sorted-set view of the keys contained in this map.
 *  <p>Note that this specification strengthens the one given in the
 *  corresponding type-specific unsorted map.
 *
 * @return a sorted-set view of the keys contained in this map.
 * @see SortedMap#keySet()
 */

@Override
SORTED_SET KEY_GENERIC keySet();

/** Returns a type-specific set view of the values contained in this map.
 * <p>Note that this specification strengthens the one given in {@link Map#values()},
 * which was already strengthened in the corresponding type-specific class,
 * but was weakened by the fact that this interface extends {@link SortedMap}.
 *
 * @return a set view of the values contained in this map.
 * @see SortedMap#values()
 */

@Override
VALUE_COLLECTION VALUE_GENERIC values();

/** Returns
the comparator associated with this sorted set, or null if it uses its keys' natural ordering.
 *
 *  <p>Note that this specification strengthens the one given in {@link SortedMap#comparator()}.
 *
 * @see SortedMap#comparator()
 */

@Override
KEY_COMPARATOR KEY_SUPER_GENERIC comparator();
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/SortedMap.drv
No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import it.unimi.dsi.fastutil.Function;

/** A type-specific {@link Function}; provides some additional methods that use polymorphism to avoid
(un)boxing.
*
* <p>Type-specific versions of {@code get()}, {@code put()} and
* {@code remove()} cannot rely on {@code null} to denote absence of
* a key.
 Rather, they return a {@linkplain #defaultReturnValue() default
* return value}, which is set to 0/false at creation, but can be changed using
* the {@code defaultReturnValue()} method.
*
* <p>For uniformity reasons, even functions returning objects implement the default
* return value (of course, in this case the default return value is
* initialized to {@code null}).
*
* <p>The default implementation of optional operations just throw an {@link
* UnsupportedOperationException}, except for the type-specific {@code
* containsKey()}, which return true. Generic versions of accessors delegate to
* the corresponding type-specific counterparts following the interface rules.
*
* <p><strong>Warning:</strong> to fall in line as much as possible with the
* {@linkplain java.util.Map standard map interface}, it is required that
* standard versions of {@code get()}, {@code put()} and
* {@code remove()} for maps with primitive-type keys or values <em>return
* {@code null} to denote
 missing keys </em> rather than wrap the default
* return value in an object. In case both keys and values are reference
* types, the default return value must be returned instead, thus violating
* the {@linkplain java.util.Map standard map interface} when the default
```

```
* return value is not {@code null}.
*
* @see Function
*/

@FunctionalInterface
#ifdef JDK_PRIMITIVE_FUNCTION
public interface FUNCTION KEY_VALUE_GENERIC extends Function<KEY_GENERIC_CLASS,
VALUE_GENERIC_CLASS>, JDK_PRIMITIVE_FUNCTION KEY_GENERIC VALUE_GENERIC {
#else
public interface FUNCTION KEY_VALUE_GENERIC extends Function<KEY_GENERIC_CLASS,
VALUE_GENERIC_CLASS> {
#endif

#ifdef JDK_PRIMITIVE_FUNCTION
#if KEY_WIDENED
 /** {@inheritDoc}
  *
  * <p>In this default implementation, the key gets narrowed down to the
  * actual key type, throwing an exception if the given key can't be
  * represented in the restricted domain. This is done for interoperability
  * with the Java 8 function environment. Its use is discouraged,
 as
  * unexpected errors can occur. Instead, the corresponding classes should be
  * used (e.g., {@link it.unimi.dsi.fastutil.ints.Int2IntFunction} instead of
  * {@link it.unimi.dsi.fastutil.shorts.Short2IntFunction}).
  *
  * @throws IllegalArgumentException If the given operand is not an element of the key domain.
  * @since 8.0.0
  * @deprecated Please use primitive types which don't have to be widened as keys.
  */
 @Deprecated
#else
 /** {@inheritDoc}
  * @since 8.0.0
  */
#endif
 @Override
 default VALUE_GENERIC_TYPE_WIDENED
JDK_PRIMITIVE_FUNCTION_APPLY(KEY_GENERIC_TYPE_WIDENED operand) { return
GET_VALUE(KEY_NARROWING(operand)); }
#endif

 /** Adds a pair to the map (optional operation).
  *
  * @param key the key.
  * @param value the value.
  * @return the old value, or the {@linkplain #defaultReturnValue() default return value} if no value was present for
```

```
the given key.
  * @see Function#put(Object,Object)
  */

 default VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE key,
 final VALUE_GENERIC_TYPE value) {
  throw new UnsupportedOperationException();
 }

 /** Returns the value to which the given key is mapped.
  *
  * @param key the key.
  * @return the corresponding value, or the {@linkplain #defaultReturnValue() default return value} if no value was
present for the given key.
  * @see Function#get(Object)
  */

 VALUE_GENERIC_TYPE GET_VALUE(KEY_TYPE key);

 /** Removes the mapping with the given key (optional operation).
  * @param key the key.
  * @return the old value, or the {@linkplain #defaultReturnValue() default return value} if no value was present for
the given key.
  * @see Function#remove(Object)
  */

 default VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE key) {
  throw new UnsupportedOperationException();
 }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default VALUE_GENERIC_CLASS put(final KEY_GENERIC_CLASS key, final
 VALUE_GENERIC_CLASS value) {
  final KEY_GENERIC_TYPE k = KEY_CLASS2TYPE(key);
  final boolean containsKey = containsKey(k);
  final VALUE_GENERIC_TYPE v = put(k, VALUE_CLASS2TYPE(value));
  return containsKey ? VALUE2OBJ(v) : null;
 }

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
```

```
 default VALUE_GENERIC_CLASS get(final Object key) {
#if KEYS_PRIMITIVE
  if (key == null) return null;
#endif
  final KEY_TYPE k = KEY_OBJ2TYPE(key);
  final VALUE_GENERIC_TYPE v = GET_VALUE(k);
  return (v != defaultReturnValue() || containsKey(k)) ? VALUE2OBJ(v) : null;
 }

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default VALUE_GENERIC_CLASS remove(final Object key) {
#if KEYS_PRIMITIVE
  if (key == null) return null;
#endif
  final KEY_TYPE k = KEY_OBJ2TYPE(key);
  return containsKey(k) ? VALUE2OBJ(REMOVE_VALUE(k)) : null;
 }

#if KEYS_PRIMITIVE

 /**
 Returns true if this function contains a mapping for the specified key.
  *
  * <p>Note that for some kind of functions (e.g., hashes) this method will
  * always return true. In particular, this default implementation always
  * returns true.
  *
  * @param key the key.
  * @return true if this function associates a value to {@code key}.
  * @see Function#containsKey(Object)
  */
 default boolean containsKey(KEY_TYPE key) {
  return true;
 }

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */

 @Deprecated
 @Override
 default boolean containsKey(final Object key) {
  return key == null ? false : containsKey(KEY_OBJ2TYPE(key));
 }

#endif
```

```
#endif

 /** Sets the default return value (optional operation).
  *
  * This value must be returned by type-specific versions of
  * {@code get()}, {@code put()} and {@code remove()} to
  * denote that the map does not contain the specified key. It must be
  * 0/{@code false}/{@code null} by default.

 *
  * @param rv the new default return value.
  * @see #defaultReturnValue()
  */

 default void defaultReturnValue(VALUE_GENERIC_TYPE rv) {
  throw new UnsupportedOperationException();
 }


 /** Gets the default return value.
  *
  * <p>This default implementation just return the default null value
  * of the type ({@code null} for objects, 0 for scalars, false for Booleans).
  *
  * @return the current default return value.
  */

 default VALUE_GENERIC_TYPE defaultReturnValue() {
  return VALUE_NULL;
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Function.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*
*
*
* For the sorting and binary search code:
*
* Copyright (C) 1999 CERN - European Organization for Nuclear Research.
*
*   Permission to use, copy, modify, distribute and sell this software and
*   its documentation for any purpose is hereby granted without fee,
*   provided that the above copyright notice appear
 in all copies and that
*   both that copyright notice and this permission notice appear in
*   supporting documentation. CERN makes no representations about the
*   suitability of this software for any purpose. It is provided "as is"
*   without expressed or implied warranty.
*/

package PACKAGE;

import it.unimi.dsi.fastutil.Arrays;
import it.unimi.dsi.fastutil.Hash;
import java.util.Random;
import java.util.concurrent.ForkJoinPool;
import java.util.concurrent.RecursiveAction;

#if ! KEY_CLASS_Integer
import it.unimi.dsi.fastutil.ints.IntArrays;
#endif

#if KEYS_PRIMITIVE

#if ! KEY_CLASS_Boolean
import java.util.concurrent.ExecutorCompletionService;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.atomic.AtomicInteger;
#endif

/** A class providing static methods and objects that do useful things with type-specific arrays.
*
* <p>In particular, the {@code forceCapacity()},
 {@code ensureCapacity()}, {@code grow()},
* {@code trim()} and {@code setLength()} methods allow to handle
```

```
* arrays much like array lists. This can be very useful when efficiency (or
* syntactic simplicity) reasons make array lists unsuitable.
*
* <p>Note that {@link it.unimi.dsi.fastutil.io.BinIO} and {@link it.unimi.dsi.fastutil.io.TextIO}
* contain several methods make it possible to load and save arrays of primitive types as sequences
* of elements in {@link java.io.DataInput} format (i.e., not as objects) or as sequences of lines of text.
*
* <h2>Sorting</h2>
*
* <p>There are several sorting methods available. The main theme is that of letting you choose
* the sorting algorithm you prefer (i.e., trading stability of mergesort for no memory allocation in quicksort).
* Several algorithms provide a parallel version, that will use the {@linkplain Runtime#availableProcessors()
number of cores available}.
* Some algorithms also provide an explicit <em>indirect</em>
 sorting facility, which makes it possible
* to sort an array using the values in another array as comparator.
*
* <p>However, if you wish to let the implementation choose an algorithm for you, both
* {@link #stableSort} and {@link #unstableSort} methods are available, which dynamically
* chooses an algorithm based on unspecified criteria (but most likely stability, array size, and array element type).
*
* <p>All comparison-based algorithm have an implementation based on a type-specific comparator.
*
* <p>As a general rule, sequential radix sort is significantly faster than quicksort or mergesort, in particular
* on random-looking data. In
* the parallel case, up to a few cores parallel radix sort is still the fastest, but at some point quicksort
* exploits parallelism better.
*
* <p>If you are fine with not knowing exactly which algorithm will be run (in particular, not knowing exactly
whether a support array will be allocated),
* the dual-pivot parallel sorts in {@link
 java.util.Arrays}
* are about 50% faster than the classical single-pivot implementation used here.
*
* <p>In any case, if sorting time is important I suggest that you benchmark your sorting load
* with your data distribution and on your architecture.
*
* @see java.util.Arrays
*/

public final class ARRAYS {

#else

import java.util.Comparator;

/** A class providing static methods and objects that do useful things with type-specific arrays.
*
```

```
* In particular, the {@code ensureCapacity()}, {@code grow()},
* {@code trim()} and {@code setLength()} methods allow to handle
* arrays much like array lists. This can be very useful when efficiency (or
* syntactic simplicity) reasons make array lists unsuitable.
*
* <p><strong>Warning:</strong> if your array is not of type {@code Object[]},
* {@link #ensureCapacity(Object[],int,int)} and {@link #grow(Object[],int,int)}
* will use {@linkplain java.lang.reflect.Array#newInstance(Class,int) reflection}
* to preserve your array
 type. Reflection is <em>significantly slower</em> than using {@code new}.
* This phenomenon is particularly
* evident in the first growth phases of an array reallocated with doubling (or similar) logic.
*
* <h2>Sorting</h2>
*
* <p>There are several sorting methods available. The main theme is that of letting you choose
* the sorting algorithm you prefer (i.e., trading stability of mergesort for no memory allocation in quicksort).
* Several algorithms provide a parallel version, that will use the {@linkplain Runtime#availableProcessors()
number of cores available}.
*
* <p>All comparison-based algorithm have an implementation based on a type-specific comparator.
*
* <p>If you are fine with not knowing exactly which algorithm will be run (in particular, not knowing exactly
whether a support array will be allocated),
* the dual-pivot parallel sorts in {@link java.util.Arrays}
* are about 50% faster than the classical single-pivot implementation used here.
*
* <p>In any case,
 if sorting time is important I suggest that you benchmark your sorting load
* with your data distribution and on your architecture.
*
* @see java.util.Arrays
*/

public final class ARRAYS {

#endif
 private ARRAYS() {}

 /** A static, final, empty array. */
 public static final KEY_TYPE[] EMPTY_ARRAY = {};

 /** A static, final, empty array to be used as default array in allocations. An
   * object distinct from {@link #EMPTY_ARRAY} makes it possible to have different
   * behaviors depending on whether the user required an empty allocation, or we are
   * just lazily delaying allocation.
   *
   * @see java.util.ArrayList
```

```
  */
 public static final KEY_TYPE[] DEFAULT_EMPTY_ARRAY = {};


#if KEY_CLASS_Object
 /** Creates a new array using a the given one as prototype.
  *
  * <p>This method returns a new array of the given length whose element
  * are of the same class as of those of {@code prototype}. In case
  * of an empty array, it tries to return {@link #EMPTY_ARRAY}, if possible.

 *
  * @param prototype an array that will be used to type the new one.
  * @param length the length of the new array.
  * @return a new array of given type and length.
  */

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 private static <K> K[] newArray(final K[] prototype, final int length) {
  final Class<?> klass = prototype.getClass();
  if (klass == Object[].class) return (K[])(length == 0 ? EMPTY_ARRAY : new Object[length]);
  return (K[])java.lang.reflect.Array.newInstance(klass.getComponentType(), length);
 }
#endif

 /** Forces an array to contain the given number of entries, preserving just a part of the array.
  *
  * @param array an array.
  * @param length the new minimum length for this array.
  * @param preserve the number of elements of the array that must be preserved in case a new allocation is
necessary.
  * @return an array with {@code length} entries whose first {@code preserve}
  * entries are the same as those of {@code array}.
  */
 public static KEY_GENERIC KEY_GENERIC_TYPE[]
 forceCapacity(final KEY_GENERIC_TYPE[] array, final int length, final int preserve) {
  final KEY_GENERIC_TYPE t[] =
#if KEY_CLASS_Object
   newArray(array, length);
#else
   new KEY_TYPE[length];
#endif
  System.arraycopy(array, 0, t, 0, preserve);
  return t;
 }

 /** Ensures that an array can contain the given number of entries.
  *
```

```
 * <p>If you cannot foresee whether this array will need again to be
 * enlarged, you should probably use {@code grow()} instead.
 *
 * @param array an array.
 * @param length the new minimum length for this array.
 * @return {@code array}, if it contains {@code length} entries or more; otherwise,
 * an array with {@code length} entries whose first {@code array.length}
 * entries are the same as those of {@code array}.
 */
 public static KEY_GENERIC KEY_GENERIC_TYPE[] ensureCapacity(final KEY_GENERIC_TYPE[] array,
final int length) {
  return ensureCapacity(array, length, array.length);
 }

 /** Ensures that an array can contain the
 given number of entries, preserving just a part of the array.
 *
 * @param array an array.
 * @param length the new minimum length for this array.
 * @param preserve the number of elements of the array that must be preserved in case a new allocation is
necessary.
 * @return {@code array}, if it can contain {@code length} entries or more; otherwise,
 * an array with {@code length} entries whose first {@code preserve}
 * entries are the same as those of {@code array}.
 */
 public static KEY_GENERIC KEY_GENERIC_TYPE[] ensureCapacity(final KEY_GENERIC_TYPE[] array,
final int length, final int preserve) {
  return length > array.length ? forceCapacity(array, length, preserve) : array;
 }

 /** Grows the given array to the maximum between the given length and
 * the current length increased by 50%, provided that the given
 * length is larger than the current length.
 *
 * <p>If you want complete control on the array growth, you
 * should probably use {@code ensureCapacity()} instead.

 *
 * @param array an array.
 * @param length the new minimum length for this array.
 * @return {@code array}, if it can contain {@code length}
 * entries; otherwise, an array with
 * max({@code length},{@code array.length}/&phi;) entries whose first
 * {@code array.length} entries are the same as those of {@code array}.
 * */

 public static KEY_GENERIC KEY_GENERIC_TYPE[] grow(final KEY_GENERIC_TYPE[] array, final int
length) {
```

```
 return grow(array, length, array.length);
 }

 /** Grows the given array to the maximum between the given length and
  * the current length increased by 50%, provided that the given
  * length is larger than the current length, preserving just a part of the array.
  *
  * <p>If you want complete control on the array growth, you
  * should probably use {@code ensureCapacity()} instead.
  *
  * @param array an array.
  * @param length the new minimum length for this array.
  * @param preserve the number of elements of the array that must be preserved
 in case a new allocation is necessary.
  * @return {@code array}, if it can contain {@code length}
  * entries; otherwise, an array with
  * max({@code length},{@code array.length}/&phi;) entries whose first
  * {@code preserve} entries are the same as those of {@code array}.
  * */

 public static KEY_GENERIC KEY_GENERIC_TYPE[] grow(final KEY_GENERIC_TYPE[] array, final int
length, final int preserve) {
  if (length > array.length) {
   final int newLength = (int)Math.max(Math.min((long)array.length + (array.length >> 1),
Arrays.MAX_ARRAY_SIZE), length);

   final KEY_GENERIC_TYPE t[] =
#if KEY_CLASS_Object
    newArray(array, newLength);
#else
    new KEY_TYPE[newLength];
#endif
   System.arraycopy(array, 0, t, 0, preserve);

   return t;
  }
  return array;

 }

 /** Trims the given array to the given length.
  *
  * @param array an array.
  * @param length the new maximum length for the array.
  * @return {@code array}, if it contains {@code length}
  * entries or less; otherwise,
 an array with
  * {@code length} entries whose entries are the same as
```

```
 * the first {@code length} entries of {@code array}.
 *
 */

 public static KEY_GENERIC KEY_GENERIC_TYPE[] trim(final KEY_GENERIC_TYPE[] array, final int length)
{
  if (length >= array.length) return array;
  final KEY_GENERIC_TYPE t[] =
#if KEY_CLASS_Object
   newArray(array, length);
#else
   length == 0 ? EMPTY_ARRAY : new KEY_TYPE[length];
#endif
  System.arraycopy(array, 0, t, 0, length);
  return t;
 }

 /** Sets the length of the given array.
 *
 * @param array an array.
 * @param length the new length for the array.
 * @return {@code array}, if it contains exactly {@code length}
 * entries; otherwise, if it contains <em>more</em> than
 * {@code length} entries, an array with {@code length} entries
 * whose entries are the same as the first {@code length} entries of
 * {@code array}; otherwise, an array with {@code length} entries
 * whose first {@code array.length} entries are the same as those
 of
 * {@code array}.
 *
 */

 public static KEY_GENERIC KEY_GENERIC_TYPE[] setLength(final KEY_GENERIC_TYPE[] array, final int
length) {
  if (length == array.length) return array;
  if (length < array.length) return trim(array, length);
  return ensureCapacity(array, length);
 }

 /** Returns a copy of a portion of an array.
 *
 * @param array an array.
 * @param offset the first element to copy.
 * @param length the number of elements to copy.
 * @return a new array containing {@code length} elements of {@code array} starting at {@code offset}.
 */

 public static KEY_GENERIC KEY_GENERIC_TYPE[] copy(final KEY_GENERIC_TYPE[] array, final int offset,
```

```
final int length) {
  ensureOffsetLength(array, offset, length);
  final KEY_GENERIC_TYPE[] a =
#if KEY_CLASS_Object
   newArray(array, length);
#else
   length == 0 ? EMPTY_ARRAY : new KEY_TYPE[length];
#endif
  System.arraycopy(array, offset, a, 0, length);
  return a;
 }

 /** Returns a copy of an array.
  *
  * @param array
 an array.
  * @return a copy of {@code array}.
  */

 public static KEY_GENERIC KEY_GENERIC_TYPE[] copy(final KEY_GENERIC_TYPE[] array) {
  return array.clone();
 }

 /** Fills the given array with the given value.
  *
  * @param array an array.
  * @param value the new value for all elements of the array.
  * @deprecated Please use the corresponding {@link java.util.Arrays} method.
  */

 @Deprecated
 public static KEY_GENERIC void fill(final KEY_GENERIC_TYPE[] array, final KEY_GENERIC_TYPE value) {
  int i = array.length;
  while(i-- != 0) array[i] = value;
 }

 /** Fills a portion of the given array with the given value.
  *
  * @param array an array.
  * @param from the starting index of the portion to fill (inclusive).
  * @param to the end index of the portion to fill (exclusive).
  * @param value the new value for all elements of the specified portion of the array.
  * @deprecated Please use the corresponding {@link java.util.Arrays} method.
  */

 @Deprecated
 public static
```

```
 KEY_GENERIC void fill(final KEY_GENERIC_TYPE[] array, final int from, int to, final KEY_GENERIC_TYPE
value) {
  ensureFromTo(array, from, to);
  if (from == 0) while(to-- != 0) array[to] = value;
  else for(int i = from; i < to; i++) array[i] = value;
 }



 /** Returns true if the two arrays are elementwise equal.
  *
  * @param a1 an array.
  * @param a2 another array.
  * @return true if the two arrays are of the same length, and their elements are equal.
  * @deprecated Please use the corresponding {@link java.util.Arrays} method, which is intrinsified in recent JVMs.
  */

 @Deprecated
 public static KEY_GENERIC boolean equals(final KEY_GENERIC_TYPE[] a1, final KEY_GENERIC_TYPE
a2[]) {
  int i = a1.length;
  if (i != a2.length) return false;
  while(i-- != 0) if (! KEY_EQUALS(a1[i], a2[i])) return false;
  return true;
 }




 /** Ensures that a range given by its first (inclusive) and last (exclusive) elements fits an array.
  *
  * <p>This method may be used whenever an array range
 check is needed.
  *
  * @param a an array.
  * @param from a start index (inclusive).
  * @param to an end index (exclusive).
  * @throws IllegalArgumentException if {@code from} is greater than {@code to}.
  * @throws ArrayIndexOutOfBoundsException if {@code from} or {@code to} are greater than the array length or
negative.
  */
 public static KEY_GENERIC void ensureFromTo(final KEY_GENERIC_TYPE[] a, final int from, final int to) {
  Arrays.ensureFromTo(a.length, from, to);
 }

 /** Ensures that a range given by an offset and a length fits an array.
  *
  * <p>This method may be used whenever an array range check is needed.
```

```
 *
 * @param a an array.
 * @param offset a start index.
 * @param length a length (the number of elements in the range).
 * @throws IllegalArgumentException if {@code length} is negative.
 * @throws ArrayIndexOutOfBoundsException if {@code offset} is negative or {@code offset}+{@code length} is
greater than the array length.
 */
public static KEY_GENERIC
void ensureOffsetLength(final KEY_GENERIC_TYPE[] a, final int offset, final int length) {
 Arrays.ensureOffsetLength(a.length, offset, length);
}

/** Ensures that two arrays are of the same length.
 *
 * @param a an array.
 * @param b another array.
 * @throws IllegalArgumentException if the two argument arrays are not of the same length.
 */
public static KEY_GENERIC void ensureSameLength(final KEY_GENERIC_TYPE[] a, final
KEY_GENERIC_TYPE[] b) {
 if (a.length != b.length) throw new IllegalArgumentException("Array size mismatch: " + a.length + " != " +
b.length);
}

private static final int QUICKSORT_NO_REC = 16;
private static final int PARALLEL_QUICKSORT_NO_FORK = 8192;
private static final int QUICKSORT_MEDIAN_OF_9 = 128;
private static final int MERGESORT_NO_REC = 16;

/** Swaps two elements of an anrray.
 *
 * @param x an array.
 * @param a a position in {@code x}.
 * @param b another position in {@code x}.
 */
public static KEY_GENERIC void swap(final KEY_GENERIC_TYPE
x[], final int a, final int b) {
 final KEY_GENERIC_TYPE t = x[a];
 x[a] = x[b];
 x[b] = t;
}

/** Swaps two sequences of elements of an array.
 *
 * @param x an array.
 * @param a a position in {@code x}.
 * @param b another position in {@code x}.
```

```
 * @param n the number of elements to exchange starting at {@code a} and {@code b}.
 */
 public static KEY_GENERIC void swap(final KEY_GENERIC_TYPE[] x, int a, int b, final int n) {
  for(int i = 0; i < n; i++, a++, b++) swap(x, a, b);
 }

 private static KEY_GENERIC int med3(final KEY_GENERIC_TYPE x[], final int a, final int b, final int c,
KEY_COMPARATOR KEY_GENERIC comp) {
  final int ab = comp.compare(x[a], x[b]);
  final int ac = comp.compare(x[a], x[c]);
  final int bc = comp.compare(x[b], x[c]);
  return (ab < 0 ?
   (bc < 0 ? b : ac < 0 ? c : a) :
   (bc > 0 ? b : ac > 0 ? c : a));
 }

 private static KEY_GENERIC void selectionSort(final KEY_GENERIC_TYPE[] a, final int from, final int to, final
KEY_COMPARATOR
 KEY_GENERIC comp) {
  for(int i = from; i < to - 1; i++) {
   int m = i;
   for(int j = i + 1; j < to; j++) if (comp.compare(a[j], a[m]) < 0) m = j;
   if (m != i) {
    final KEY_GENERIC_TYPE u = a[i];
    a[i] = a[m];
    a[m] = u;
   }
  }
 }

 private static KEY_GENERIC void insertionSort(final KEY_GENERIC_TYPE[] a, final int from, final int to, final
KEY_COMPARATOR KEY_GENERIC comp) {
  for (int i = from; ++i < to;) {
   KEY_GENERIC_TYPE t = a[i];
   int j = i;
   for (KEY_GENERIC_TYPE u = a[j - 1]; comp.compare(t, u) < 0; u = a[--j - 1]) {
    a[j] = u;
    if (from == j - 1) {
     --j;
     break;
    }
   }
   a[j] = t;
  }
 }
```

```
/** Sorts the specified range of elements according to the order induced by the specified
 * comparator using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265,
1993.
 *
 * <p>Note that this implementation does not allocate any object, contrarily to the implementation
 * used to sort primitive types in {@link java.util.Arrays}, which switches to mergesort on large inputs.
 *
 * @param x the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 * @param comp the comparator to determine the sorting order.
 *
 */
 public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[] x, final int from, final int to, final
KEY_COMPARATOR KEY_GENERIC comp) {
  final int len = to - from;
  // Selection sort on smallest arrays
  if (len < QUICKSORT_NO_REC) {
   selectionSort(x, from, to, comp);
   return;
  }

  // Choose a partition element, v
  int m = from + len / 2;
  int l = from;
  int n = to - 1;
  if (len > QUICKSORT_MEDIAN_OF_9) {  // Big arrays, pseudomedian of 9
   int s = len / 8;
   l = med3(x, l, l + s, l + 2 * s, comp);
   m
= med3(x, m - s, m, m + s, comp);
   n = med3(x, n - 2 * s, n - s, n, comp);
  }
  m = med3(x, l, m, n, comp); // Mid-size, med of 3

  final KEY_GENERIC_TYPE v = x[m];

  // Establish Invariant: v* (<v)* (>v)* v*
  int a = from, b = a, c = to - 1, d = c;
  while(true) {
   int comparison;
   while (b <= c && (comparison = comp.compare(x[b], v)) <= 0) {
    if (comparison == 0) swap(x, a++, b);
    b++;
```

```
  }
  while (c >= b && (comparison = comp.compare(x[c], v)) >=0) {
   if (comparison == 0) swap(x, c, d--);
   c--;
  }
  if (b > c) break;
  swap(x, b++, c--);
 }

 // Swap partition elements back to middle
 int s;
 s = Math.min(a - from, b - a);
 swap(x, from, b - s, s);
 s = Math.min(d - c, to - d - 1);
 swap(x, b, to - s, s);

 // Recursively sort non-partition-elements
 if ((s = b - a) > 1) quickSort(x, from, from + s, comp);
 if ((s = d - c) > 1) quickSort(x, to - s, to, comp);
}

/** Sorts an array according to the order induced by the specified
 * comparator
using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>Note that this implementation does not allocate any object, contrarily to the implementation
 * used to sort primitive types in {@link java.util.Arrays}, which switches to mergesort on large inputs.
 *
 * @param x the array to be sorted.
 * @param comp the comparator to determine the sorting order.
 *
 */
 public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[] x, final KEY_COMPARATOR
KEY_GENERIC comp) {
 quickSort(x, 0, x.length, comp);
 }


 protected static class ForkJoinQuickSortComp KEY_GENERIC extends RecursiveAction {
  private static final long serialVersionUID = 1L;
  private final int from;
  private final int to;
  private final KEY_GENERIC_TYPE[] x;
  private final KEY_COMPARATOR KEY_GENERIC comp;
```

```
 public
 ForkJoinQuickSortComp(final KEY_GENERIC_TYPE[] x , final int from , final int to, final
KEY_COMPARATOR KEY_GENERIC comp) {
  this.from = from;
  this.to = to;
  this.x = x;
  this.comp = comp;
 }

 @Override
 protected void compute() {
  final KEY_GENERIC_TYPE[] x = this.x;
  final int len = to - from;
  if (len < PARALLEL_QUICKSORT_NO_FORK) {
   quickSort(x, from, to, comp);
   return;
  }
  // Choose a partition element, v
  int m = from + len / 2;
  int l = from;
  int n = to - 1;
  int s = len / 8;
  l = med3(x, l, l + s, l + 2 * s, comp);
  m = med3(x, m - s, m, m + s, comp);
  n = med3(x, n - 2 * s, n - s, n, comp);
  m = med3(x, l, m, n, comp);
  final KEY_GENERIC_TYPE v = x[m];
  // Establish Invariant: v* (<v)* (>v)* v*
  int a = from, b = a, c = to - 1, d = c;
  while (true) {
   int comparison;
   while (b <= c && (comparison = comp.compare(x[b], v)) <= 0) {
    if (comparison == 0) swap(x, a++, b);
    b++;
   }
   while (c >= b
&& (comparison = comp.compare(x[c], v)) >= 0) {
    if (comparison == 0) swap(x, c, d--);
    c--;
   }
   if (b > c) break;
   swap(x, b++, c--);
  }
  // Swap partition elements back to middle
  int t;
  s = Math.min(a - from, b - a);
  swap(x, from, b - s, s);
```

```
  s = Math.min(d - c, to - d - 1);
  swap(x, b, to - s, s);
  // Recursively sort non-partition-elements
  s = b - a;
  t = d - c;
  if (s > 1 && t > 1) invokeAll(new ForkJoinQuickSortComp KEY_GENERIC_DIAMOND(x, from, from + s,
comp), new ForkJoinQuickSortComp KEY_GENERIC_DIAMOND(x, to - t, to, comp));
  else if (s > 1) invokeAll(new ForkJoinQuickSortComp KEY_GENERIC_DIAMOND(x, from, from + s, comp));
  else invokeAll(new ForkJoinQuickSortComp KEY_GENERIC_DIAMOND(x, to - t, to, comp));
 }
}

/** Sorts the specified range of elements according to the order induced by the specified
 * comparator using a parallel quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and
M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This implementation uses a {@link ForkJoinPool} executor service with
 * {@link Runtime#availableProcessors()} parallel threads.
 *
 * @param x the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 * @param comp the comparator to determine the sorting order.
 */
public static KEY_GENERIC void parallelQuickSort(final KEY_GENERIC_TYPE[] x, final int from, final int to,
final KEY_COMPARATOR KEY_GENERIC comp) {
 if (to - from < PARALLEL_QUICKSORT_NO_FORK) quickSort(x, from, to, comp);
 else {
  final ForkJoinPool pool = new ForkJoinPool(Runtime.getRuntime().availableProcessors());
  pool.invoke(new ForkJoinQuickSortComp KEY_GENERIC_DIAMOND(x, from, to, comp));
  pool.shutdown();
 }
}

/** Sorts an array
according to the order induced by the specified
 * comparator using a parallel quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This implementation uses a {@link ForkJoinPool} executor service with
 * {@link Runtime#availableProcessors()} parallel threads.
```

```
 *
 * @param x the array to be sorted.
 * @param comp the comparator to determine the sorting order.
 */
 public static KEY_GENERIC void parallelQuickSort(final KEY_GENERIC_TYPE[] x, final
KEY_COMPARATOR KEY_GENERIC comp) {
  parallelQuickSort(x, 0, x.length, comp);
 }


 SUPPRESS_WARNINGS_KEY_UNCHECKED
 private static KEY_GENERIC int med3(final KEY_GENERIC_TYPE x[], final int a, final int b, final int c) {
  final int ab = KEY_CMP(x[a], x[b]);
  final int ac = KEY_CMP(x[a], x[c]);
  final int bc = KEY_CMP(x[b], x[c]);
  return
 (ab < 0 ?
   (bc < 0 ? b : ac < 0 ? c : a) :
   (bc > 0 ? b : ac > 0 ? c : a));
 }

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 private static KEY_GENERIC void selectionSort(final KEY_GENERIC_TYPE[] a, final int from, final int to) {
  for(int i = from; i < to - 1; i++) {
   int m = i;
   for(int j = i + 1; j < to; j++) if (KEY_LESS(a[j], a[m])) m = j;
   if (m != i) {
    final KEY_GENERIC_TYPE u = a[i];
    a[i] = a[m];
    a[m] = u;
   }
  }
 }

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 private static KEY_GENERIC void insertionSort(final KEY_GENERIC_TYPE[] a, final int from, final int to) {
  for (int i = from; ++i < to;) {
   KEY_GENERIC_TYPE t = a[i];
   int j = i;
   for (KEY_GENERIC_TYPE u = a[j - 1]; KEY_LESS(t, u); u = a[--j - 1]) {
    a[j] = u;
    if (from == j - 1) {
     --j;
     break;
    }
   }
   a[j] = t;
  }
```

```
}

/** Sorts the specified range of elements according to the natural ascending order using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from
Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>Note that this implementation does not allocate any object, contrarily to the implementation
 * used to sort primitive types in {@link java.util.Arrays}, which switches to mergesort on large inputs.
 *
 * @param x the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[] x, final int from, final int to) {
 final int len = to - from;
 // Selection sort on smallest arrays
 if (len < QUICKSORT_NO_REC) {
  selectionSort(x, from, to);
  return;
 }

 // Choose a partition element, v
 int m = from + len / 2;
 int l = from;
 int n = to - 1;
 if (len > QUICKSORT_MEDIAN_OF_9) {  // Big arrays,
pseudomedian of 9
  int s = len / 8;
  l = med3(x, l, l + s, l + 2 * s);
  m = med3(x, m - s, m, m + s);
  n = med3(x, n - 2 * s, n - s, n);
 }
 m = med3(x, l, m, n); // Mid-size, med of 3

 final KEY_GENERIC_TYPE v = x[m];

 // Establish Invariant: v* (<v)* (>v)* v*
 int a = from, b = a, c = to - 1, d = c;
 while(true) {
  int comparison;
  while (b <= c && (comparison = KEY_CMP(x[b], v)) <= 0) {
   if (comparison == 0) swap(x, a++, b);
   b++;
```

```
  }
  while (c >= b && (comparison = KEY_CMP(x[c], v)) >=0) {
   if (comparison == 0) swap(x, c, d--);
   c--;
  }
  if (b > c) break;
  swap(x, b++, c--);
 }

 // Swap partition elements back to middle
 int s;
 s = Math.min(a - from, b - a);
 swap(x, from, b - s, s);
 s = Math.min(d - c, to - d - 1);
 swap(x, b, to - s, s);

 // Recursively sort non-partition-elements
 if ((s = b - a) > 1) quickSort(x, from, from + s);
 if ((s = d - c) > 1) quickSort(x, to - s, to);
}

/** Sorts an array according to the natural
ascending order using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>Note that this implementation does not allocate any object, contrarily to the implementation
 * used to sort primitive types in {@link java.util.Arrays}, which switches to mergesort on large inputs.
 *
 * @param x the array to be sorted.
 *
 */
public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[] x) {
 quickSort(x, 0, x.length);
}

protected static class ForkJoinQuickSort KEY_GENERIC extends RecursiveAction {
 private static final long serialVersionUID = 1L;
 private final int from;
 private final int to;
 private final KEY_GENERIC_TYPE[] x;

 public ForkJoinQuickSort(final KEY_GENERIC_TYPE[] x , final int from , final int to) {
  this.from = from;
  this.to = to;
  this.x = x;
```

```
 }

 @Override
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 protected
void compute() {
  final KEY_GENERIC_TYPE[] x = this.x;
  final int len = to - from;
  if (len < PARALLEL_QUICKSORT_NO_FORK) {
   quickSort(x, from, to);
   return;
  }
  // Choose a partition element, v
  int m = from + len / 2;
  int l = from;
  int n = to - 1;
  int s = len / 8;
  l = med3(x, l, l + s, l + 2 * s);
  m = med3(x, m - s, m, m + s);
  n = med3(x, n - 2 * s, n - s, n);
  m = med3(x, l, m, n);
  final KEY_GENERIC_TYPE v = x[m];
  // Establish Invariant: v* (<v)* (>v)* v*
  int a = from, b = a, c = to - 1, d = c;
  while (true) {
   int comparison;
   while (b <= c && (comparison = KEY_CMP(x[b], v)) <= 0) {
    if (comparison == 0) swap(x, a++, b);
    b++;
   }
   while (c >= b && (comparison = KEY_CMP(x[c], v)) >= 0) {
    if (comparison == 0) swap(x, c, d--);
    c--;
   }
   if (b > c) break;
   swap(x, b++, c--);
  }
  // Swap partition elements back to middle
  int t;
  s
= Math.min(a - from, b - a);
  swap(x, from, b - s, s);
  s = Math.min(d - c, to - d - 1);
  swap(x, b, to - s, s);
  // Recursively sort non-partition-elements
  s = b - a;
  t = d - c;
  if (s > 1 && t > 1) invokeAll(new ForkJoinQuickSort KEY_GENERIC_DIAMOND(x, from, from + s), new
```

```
ForkJoinQuickSort KEY_GENERIC_DIAMOND(x, to - t, to));
  else if (s > 1) invokeAll(new ForkJoinQuickSort KEY_GENERIC_DIAMOND(x, from, from + s));
  else invokeAll(new ForkJoinQuickSort KEY_GENERIC_DIAMOND(x, to - t, to));
 }
}

/**  Sorts the specified range of elements according to the natural ascending order using a parallel quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This implementation uses a {@link ForkJoinPool} executor service with
 * {@link Runtime#availableProcessors()} parallel
threads.
 *
 * @param x the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
public static KEY_GENERIC void parallelQuickSort(final KEY_GENERIC_TYPE[] x, final int from, final int to) {
 if (to - from < PARALLEL_QUICKSORT_NO_FORK) quickSort(x, from, to);
 else {
  final ForkJoinPool pool = new ForkJoinPool(Runtime.getRuntime().availableProcessors());
  pool.invoke(new ForkJoinQuickSort KEY_GENERIC_DIAMOND(x, from, to));
  pool.shutdown();
 }
}

/** Sorts an array according to the natural ascending order using a parallel quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This implementation uses a {@link ForkJoinPool} executor service with
 * {@link
Runtime#availableProcessors()} parallel threads.
 *
 * @param x the array to be sorted.
 *
 */
public static KEY_GENERIC void parallelQuickSort(final KEY_GENERIC_TYPE[] x) {
 parallelQuickSort(x, 0, x.length);
}


SUPPRESS_WARNINGS_KEY_UNCHECKED
```

```
 private static KEY_GENERIC int med3Indirect(final int perm[], final KEY_GENERIC_TYPE x[], final int a, final
int b, final int c) {
  final KEY_GENERIC_TYPE aa = x[perm[a]];
  final KEY_GENERIC_TYPE bb = x[perm[b]];
  final KEY_GENERIC_TYPE cc = x[perm[c]];
  final int ab = KEY_CMP(aa, bb);
  final int ac = KEY_CMP(aa, cc);
  final int bc = KEY_CMP(bb, cc);
  return (ab < 0 ?
   (bc < 0 ? b : ac < 0 ? c : a) :
   (bc > 0 ? b : ac > 0 ? c : a));
 }

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 private static KEY_GENERIC void insertionSortIndirect(final int[] perm, final KEY_GENERIC_TYPE[] a, final int
from, final int to) {
  for (int i = from; ++i < to;) {
   int t = perm[i];
   int j = i;
   for (int u = perm[j - 1]; KEY_LESS(a[t], a[u]); u = perm[--j
 - 1]) {
    perm[j] = u;
    if (from == j - 1) {
     --j;
     break;
    }
   }
   perm[j] = t;
  }
 }

 /** Sorts the specified range of elements according to the natural ascending order using indirect quicksort.
  *
  * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
  * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
  * 1249&minus;1265, 1993.
  *
  * <p>This method implement an <em>indirect</em> sort. The elements of {@code perm} (which must
  * be exactly the numbers in the interval {@code [0..perm.length)}) will be permuted so that
  * {@code x[perm[i]] &le; x[perm[i + 1]]}.
  *
  * <p>Note that this implementation does not allocate any object, contrarily to the implementation
  * used to sort primitive types in {@link java.util.Arrays}, which switches to mergesort on large inputs.
  *
  * @param perm a permutation array indexing {@code x}.
  * @param x the array to be sorted.
```

```
* @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC void quickSortIndirect(final int[] perm, final KEY_GENERIC_TYPE[] x, final int
from, final int to) {
 final int len = to - from;
 // Selection sort on smallest arrays
 if (len < QUICKSORT_NO_REC) {
  insertionSortIndirect(perm, x, from, to);
  return;
 }

 // Choose a partition element, v
 int m = from + len / 2;
 int l = from;
 int n = to - 1;
 if (len > QUICKSORT_MEDIAN_OF_9) {  // Big arrays, pseudomedian of 9
  int s = len / 8;
  l = med3Indirect(perm, x, l, l + s, l + 2 * s);
  m = med3Indirect(perm, x, m - s, m, m + s);
  n = med3Indirect(perm, x, n - 2 * s, n - s, n);
 }
 m = med3Indirect(perm, x, l, m, n); // Mid-size, med of 3

 final KEY_GENERIC_TYPE v = x[perm[m]];

 // Establish Invariant: v* (<v)* (>v)* v*
 int a = from, b = a, c = to - 1, d
= c;
 while(true) {
  int comparison;
  while (b <= c && (comparison = KEY_CMP(x[perm[b]], v)) <= 0) {
   if (comparison == 0) IntArrays.swap(perm, a++, b);
   b++;
  }
  while (c >= b && (comparison = KEY_CMP(x[perm[c]], v)) >=0) {
   if (comparison == 0) IntArrays.swap(perm, c, d--);
   c--;
  }
  if (b > c) break;
  IntArrays.swap(perm, b++, c--);
 }

 // Swap partition elements back to middle
 int s;
 s = Math.min(a - from, b - a);
```

```
 IntArrays.swap(perm, from, b - s, s);
 s = Math.min(d - c, to - d - 1);
 IntArrays.swap(perm, b, to - s, s);

 // Recursively sort non-partition-elements
 if ((s = b - a) > 1) quickSortIndirect(perm, x, from, from + s);
 if ((s = d - c) > 1) quickSortIndirect(perm, x, to - s, to);
}

/** Sorts an array according to the natural ascending order using indirect quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software:
Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This method implement an <em>indirect</em> sort. The elements of {@code perm} (which must
 * be exactly the numbers in the interval {@code [0..perm.length)}) will be permuted so that
 * {@code x[perm[i]] &le; x[perm[i + 1]]}.
 *
 * <p>Note that this implementation does not allocate any object, contrarily to the implementation
 * used to sort primitive types in {@link java.util.Arrays}, which switches to mergesort on large inputs.
 *
 * @param perm a permutation array indexing {@code x}.
 * @param x the array to be sorted.
 */
public static KEY_GENERIC void quickSortIndirect(final int perm[], final KEY_GENERIC_TYPE[] x) {
 quickSortIndirect(perm, x, 0, x.length);
}

protected static class ForkJoinQuickSortIndirect KEY_GENERIC extends RecursiveAction {
 private static final long serialVersionUID = 1L;
 private final int from;
 private final int to;
 private final int[] perm;
 private final
KEY_GENERIC_TYPE[] x;

 public ForkJoinQuickSortIndirect(final int perm[], final KEY_GENERIC_TYPE[] x , final int from , final int to) {
  this.from = from;
  this.to = to;
  this.x = x;
  this.perm = perm;
 }

 @Override
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 protected void compute() {
```

```
  final KEY_GENERIC_TYPE[] x = this.x;
  final int len = to - from;
  if (len < PARALLEL_QUICKSORT_NO_FORK) {
   quickSortIndirect(perm, x, from, to);
   return;
  }
  // Choose a partition element, v
  int m = from + len / 2;
  int l = from;
  int n = to - 1;
  int s = len / 8;
  l = med3Indirect(perm, x, l, l + s, l + 2 * s);
  m = med3Indirect(perm, x, m - s, m, m + s);
  n = med3Indirect(perm, x, n - 2 * s, n - s, n);
  m = med3Indirect(perm, x, l, m, n);
  final KEY_GENERIC_TYPE v = x[perm[m]];
  // Establish Invariant: v* (<v)* (>v)* v*
  int a = from, b = a, c = to - 1, d = c;
  while (true) {
   int comparison;
   while (b <= c && (comparison = KEY_CMP(x[perm[b]], v))
<= 0) {
    if (comparison == 0) IntArrays.swap(perm, a++, b);
    b++;
   }
   while (c >= b && (comparison = KEY_CMP(x[perm[c]], v)) >= 0) {
    if (comparison == 0) IntArrays.swap(perm, c, d--);
    c--;
   }
   if (b > c) break;
   IntArrays.swap(perm, b++, c--);
  }
  // Swap partition elements back to middle
  int t;
  s = Math.min(a - from, b - a);
  IntArrays.swap(perm, from, b - s, s);
  s = Math.min(d - c, to - d - 1);
  IntArrays.swap(perm, b, to - s, s);
  // Recursively sort non-partition-elements
  s = b - a;
  t = d - c;
  if (s > 1 && t > 1) invokeAll(new ForkJoinQuickSortIndirect KEY_GENERIC_DIAMOND(perm, x, from, from +
s), new ForkJoinQuickSortIndirect KEY_GENERIC_DIAMOND(perm, x, to - t, to));
  else if (s > 1) invokeAll(new ForkJoinQuickSortIndirect KEY_GENERIC_DIAMOND(perm, x, from, from + s));
  else invokeAll(new ForkJoinQuickSortIndirect KEY_GENERIC_DIAMOND(perm, x, to - t, to));
 }
}
```

```
/**  Sorts the specified range of elements according
to the natural ascending order using a parallel indirect quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This method implement an <em>indirect</em> sort. The elements of {@code perm} (which must
 * be exactly the numbers in the interval {@code [0..perm.length)}) will be permuted so that
 * {@code x[perm[i]] &le; x[perm[i + 1]]}.
 *
 * <p>This implementation uses a {@link ForkJoinPool} executor service with
 * {@link Runtime#availableProcessors()} parallel threads.
 *
 * @param perm a permutation array indexing {@code x}.
 * @param x the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
public static KEY_GENERIC void parallelQuickSortIndirect(final int[]
perm, final KEY_GENERIC_TYPE[] x, final int from, final int to) {
 if (to - from < PARALLEL_QUICKSORT_NO_FORK) quickSortIndirect(perm, x, from, to);
 else {
  final ForkJoinPool pool = new ForkJoinPool(Runtime.getRuntime().availableProcessors());
  pool.invoke(new ForkJoinQuickSortIndirect KEY_GENERIC_DIAMOND(perm, x, from, to));
  pool.shutdown();
 }
}

/** Sorts an array according to the natural ascending order using a parallel indirect quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This method implement an <em>indirect</em> sort. The elements of {@code perm} (which must
 * be exactly the numbers in the interval {@code [0..perm.length)}) will be permuted so that
 * {@code x[perm[i]] &le; x[perm[i + 1]]}.
 *
 * <p>This implementation uses a {@link ForkJoinPool} executor
service with
 * {@link Runtime#availableProcessors()} parallel threads.
 *
 * @param perm a permutation array indexing {@code x}.
 * @param x the array to be sorted.
 *
 */
public static KEY_GENERIC void parallelQuickSortIndirect(final int perm[], final KEY_GENERIC_TYPE[] x) {
```

```
 parallelQuickSortIndirect(perm, x, 0, x.length);
 }

 /** Stabilizes a permutation.
  *
  * <p>This method can be used to stabilize the permutation generated by an indirect sorting, assuming that
  * initially the permutation array was in ascending order (e.g., the identity, as usually happens). This method
  * scans the permutation, and for each non-singleton block of elements with the same associated values in {@code
x},
  * permutes them in ascending order. The resulting permutation corresponds to a stable sort.
  *
  * <p>Usually combining an unstable indirect sort and this method is more efficient than using a stable sort,
  * as most stable sort algorithms require a support array.
  *
  * <p>More
 precisely, assuming that {@code x[perm[i]] &le; x[perm[i + 1]]}, after
  * stabilization we will also have that {@code x[perm[i]] = x[perm[i + 1]]} implies
  * {@code perm[i] &le; perm[i + 1]}.
  *
  * @param perm a permutation array indexing {@code x} so that it is sorted.
  * @param x the sorted array to be stabilized.
  * @param from the index of the first element (inclusive) to be stabilized.
  * @param to the index of the last element (exclusive) to be stabilized.
  */
 public static KEY_GENERIC void stabilize(final int perm[], final KEY_GENERIC_TYPE[] x, final int from, final
int to) {
  int curr = from;
  for(int i = from + 1; i < to; i++) {
   if (x[perm[i]] != x[perm[curr]]) {
    if (i - curr > 1) IntArrays.parallelQuickSort(perm, curr, i);
    curr = i;
   }
  }
  if (to - curr > 1) IntArrays.parallelQuickSort(perm, curr, to);
 }

 /** Stabilizes a permutation.
  *
  * <p>This method can be used to stabilize the permutation generated by an indirect sorting, assuming that

 * initially the permutation array was in ascending order (e.g., the identity, as usually happens). This method
  * scans the permutation, and for each non-singleton block of elements with the same associated values in {@code
x},
  * permutes them in ascending order. The resulting permutation corresponds to a stable sort.
  *
  * <p>Usually combining an unstable indirect sort and this method is more efficient than using a stable sort,
  * as most stable sort algorithms require a support array.
  *
```

```
 * <p>More precisely, assuming that {@code x[perm[i]] &le; x[perm[i + 1]]}, after
 * stabilization we will also have that {@code x[perm[i]] = x[perm[i + 1]]} implies
 * {@code perm[i] &le; perm[i + 1]}.
 *
 * @param perm a permutation array indexing {@code x} so that it is sorted.
 * @param x the sorted array to be stabilized.
 */
public static KEY_GENERIC void stabilize(final int perm[], final KEY_GENERIC_TYPE[] x) {
 stabilize(perm, x, 0, perm.length);
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private
static KEY_GENERIC int med3(final KEY_GENERIC_TYPE x[], final KEY_GENERIC_TYPE[] y, final int a,
final int b, final int c) {
 int t;
 final int ab = (t = KEY_CMP(x[a], x[b])) == 0 ? KEY_CMP(y[a], y[b]) : t;
 final int ac = (t = KEY_CMP(x[a], x[c])) == 0 ? KEY_CMP(y[a], y[c]) : t;
 final int bc = (t = KEY_CMP(x[b], x[c])) == 0 ? KEY_CMP(y[b], y[c]) : t;
 return (ab < 0 ?
   (bc < 0 ? b : ac < 0 ? c : a) :
   (bc > 0 ? b : ac > 0 ? c : a));
}

private static KEY_GENERIC void swap(final KEY_GENERIC_TYPE x[], final KEY_GENERIC_TYPE[] y, final
int a, final int b) {
 final KEY_GENERIC_TYPE t = x[a];
 final KEY_GENERIC_TYPE u = y[a];
 x[a] = x[b];
 y[a] = y[b];
 x[b] = t;
 y[b] = u;
}

private static KEY_GENERIC void swap(final KEY_GENERIC_TYPE[] x, final KEY_GENERIC_TYPE[] y, int a,
int b, final int n) {
 for (int i = 0; i < n; i++, a++, b++) swap(x, y, a, b);
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private static KEY_GENERIC void selectionSort(final KEY_GENERIC_TYPE[]
a, final KEY_GENERIC_TYPE[] b, final int from, final int to) {
 for(int i = from; i < to - 1; i++) {
  int m = i, u;
  for(int j = i + 1; j < to; j++)
   if ((u = KEY_CMP(a[j], a[m])) < 0 || u == 0 && KEY_LESS(b[j], b[m])) m = j;

  if (m != i) {
```

```
   KEY_GENERIC_TYPE t = a[i];
   a[i] = a[m];
   a[m] = t;
   t = b[i];
   b[i] = b[m];
   b[m] = t;
  }
 }
}

/** Sorts the specified range of elements of two arrays according to the natural lexicographical
 *  ascending order using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of
 * elements in the same position in the two provided arrays will be considered a single key, and
 * permuted accordingly. In the end, either {@code
x[i] &lt; x[i + 1]} or <code>x[i]
 * == x[i + 1]</code> and {@code y[i] &le; y[i + 1]}.
 *
 * @param x the first array to be sorted.
 * @param y the second array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[] x, final KEY_GENERIC_TYPE[] y,
final int from, final int to) {
 final int len = to - from;
 if (len < QUICKSORT_NO_REC) {
  selectionSort(x, y, from, to);
  return;
 }
 // Choose a partition element, v
 int m = from + len / 2;
 int l = from;
 int n = to - 1;
 if (len > QUICKSORT_MEDIAN_OF_9) { // Big arrays, pseudomedian of 9
  int s = len / 8;
  l = med3(x, y, l, l + s, l + 2 * s);
  m = med3(x, y, m - s, m, m + s);
  n = med3(x, y, n - 2 * s, n - s, n);
 }
 m = med3(x, y, l, m, n); // Mid-size, med of 3
```

```
 final KEY_GENERIC_TYPE v = x[m], w = y[m];
 //
Establish Invariant: v* (<v)* (>v)* v*
 int a = from, b = a, c = to - 1, d = c;
 while (true) {
  int comparison, t;
  while (b <= c && (comparison = (t = KEY_CMP(x[b], v)) == 0 ? KEY_CMP(y[b], w) : t) <= 0) {
   if (comparison == 0) swap(x, y, a++, b);
   b++;
  }
  while (c >= b && (comparison = (t = KEY_CMP(x[c], v)) == 0 ? KEY_CMP(y[c], w) : t) >= 0) {
   if (comparison == 0) swap(x, y, c, d--);
   c--;
  }
  if (b > c) break;
  swap(x, y, b++, c--);
 }
 // Swap partition elements back to middle
 int s;
 s = Math.min(a - from, b - a);
 swap(x, y, from, b - s, s);
 s = Math.min(d - c, to - d - 1);
 swap(x, y, b, to - s, s);
 // Recursively sort non-partition-elements
 if ((s = b - a) > 1) quickSort(x, y, from, from + s);
 if ((s = d - c) > 1) quickSort(x, y, to - s, to);
}

/** Sorts two arrays according to the natural lexicographical ascending order using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley
and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of
 * elements in the same position in the two provided arrays will be considered a single key, and
 * permuted accordingly. In the end, either {@code x[i] &lt; x[i + 1]} or <code>x[i]
 * == x[i + 1]</code> and {@code y[i] &le; y[i + 1]}.
 *
 * @param x the first array to be sorted.
 * @param y the second array to be sorted.
 */
 public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[] x, final KEY_GENERIC_TYPE[] y)
{
 ensureSameLength(x, y);
 quickSort(x, y, 0, x.length);
 }
```

```
protected static class ForkJoinQuickSort2 KEY_GENERIC extends RecursiveAction {
 private static final long serialVersionUID = 1L;
 private final int from;
 private final int to;
 private final KEY_GENERIC_TYPE[] x, y;

 public ForkJoinQuickSort2(final
KEY_GENERIC_TYPE[] x, final KEY_GENERIC_TYPE[] y, final int from , final int to) {
  this.from = from;
  this.to = to;
  this.x = x;
  this.y = y;
 }

 @Override
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 protected void compute() {
  final KEY_GENERIC_TYPE[] x = this.x;
  final KEY_GENERIC_TYPE[] y = this.y;
  final int len = to - from;
  if (len < PARALLEL_QUICKSORT_NO_FORK) {
   quickSort(x, y, from, to);
   return;
  }
  // Choose a partition element, v
  int m = from + len / 2;
  int l = from;
  int n = to - 1;
  int s = len / 8;
  l = med3(x, y, l, l + s, l + 2 * s);
  m = med3(x, y, m - s, m, m + s);
  n = med3(x, y, n - 2 * s, n - s, n);
  m = med3(x, y, l, m, n);
  final KEY_GENERIC_TYPE v = x[m], w = y[m];
  // Establish Invariant: v* (<v)* (>v)* v*
  int a = from, b = a, c = to - 1, d = c;
  while (true) {
   int comparison, t;
   while (b <= c && (comparison = (t = KEY_CMP(x[b], v)) == 0 ? KEY_CMP(y[b], w) : t) <= 0) {
    if (comparison == 0) swap(x,
y, a++, b);
    b++;
   }
   while (c >= b && (comparison = (t = KEY_CMP(x[c], v)) == 0 ? KEY_CMP(y[c], w) : t) >= 0) {
    if (comparison == 0) swap(x, y, c, d--);
    c--;
   }
```

```
   if (b > c) break;
   swap(x, y, b++, c--);
  }
  // Swap partition elements back to middle
  int t;
  s = Math.min(a - from, b - a);
  swap(x, y, from, b - s, s);
  s = Math.min(d - c, to - d - 1);
  swap(x, y, b, to - s, s);
  s = b - a;
  t = d - c;
  // Recursively sort non-partition-elements
  if (s > 1 && t > 1) invokeAll(new ForkJoinQuickSort2 KEY_GENERIC_DIAMOND(x, y, from, from + s), new
ForkJoinQuickSort2 KEY_GENERIC_DIAMOND(x, y, to - t, to));
  else if (s > 1) invokeAll(new ForkJoinQuickSort2 KEY_GENERIC_DIAMOND(x, y, from, from + s));
  else invokeAll(new ForkJoinQuickSort2 KEY_GENERIC_DIAMOND(x, y, to - t, to));
 }
}

/** Sorts the specified range of elements of two arrays according to the natural lexicographical
 *  ascending order using a parallel quicksort.
 *

* <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of
 * elements in the same position in the two provided arrays will be considered a single key, and
 * permuted accordingly. In the end, either {@code x[i] &lt; x[i + 1]} or <code>x[i]
 * == x[i + 1]</code> and {@code y[i] &le; y[i + 1]}.
 *
 * <p>This implementation uses a {@link ForkJoinPool} executor service with
 * {@link Runtime#availableProcessors()} parallel threads.
 *
 * @param x the first array to be sorted.
 * @param y the second array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
public static KEY_GENERIC void parallelQuickSort(final
KEY_GENERIC_TYPE[] x, final KEY_GENERIC_TYPE[] y, final int from, final int to) {
 if (to - from < PARALLEL_QUICKSORT_NO_FORK) quickSort(x, y, from, to);
 final ForkJoinPool pool = new ForkJoinPool(Runtime.getRuntime().availableProcessors());
 pool.invoke(new ForkJoinQuickSort2 KEY_GENERIC_DIAMOND(x, y, from, to));
 pool.shutdown();
}
```

```
 /** Sorts two arrays according to the natural lexicographical
  *  ascending order using a parallel quicksort.
  *
  * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
  * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
  * 1249&minus;1265, 1993.
  *
  * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of
  * elements in the same position in the two provided arrays will be considered a single key, and
  * permuted accordingly. In the end, either {@code x[i] &lt; x[i + 1]} or <code>x[i]
  * == x[i + 1]</code> and
 {@code y[i] &le; y[i + 1]}.
  *
  * <p>This implementation uses a {@link ForkJoinPool} executor service with
  * {@link Runtime#availableProcessors()} parallel threads.
  *
  * @param x the first array to be sorted.
  * @param y the second array to be sorted.
  */
 public static KEY_GENERIC void parallelQuickSort(final KEY_GENERIC_TYPE[] x, final
KEY_GENERIC_TYPE[] y) {
  ensureSameLength(x, y);
  parallelQuickSort(x, y, 0, x.length);
 }

 /** Sorts an array according to the natural ascending order,
  * potentially dynamically choosing an appropriate algorithm given the type and size of the array. The
  * sort will be stable unless it is provable that it would be impossible for there to be any difference
  * between a stable and unstable sort for the given type, in which case stability is meaningless and thus
  * unspecified.
  *
  * @param a the array to be sorted.
  * @param from the index of the first element (inclusive) to be sorted.
  * @param to the index of the last element (exclusive)
 to be sorted.
  * @since 8.3.0
  */
 public static KEY_GENERIC void unstableSort(final KEY_GENERIC_TYPE a[], final int from, final int to) {
#if KEYS_PRIMITIVE && !KEY_CLASS_Boolean
#if KEY_CLASS_Byte
   // JDK's sort for bytes uses bucket sort; not even radix sort can beat that.
   java.util.Arrays.sort(a, from, to);
#else
   // TODO For some TBD threshold, delegate to java.util.Arrays.sort if under it.
   if (to - from >= RADIX_SORT_MIN_THRESHOLD) {
    radixSort(a, from, to);
   } else {
    quickSort(a, from, to);
```

```
  }
#endif
#else
  quickSort(a, from, to);
#endif
 }

 /** Sorts the specified range of elements according to the natural ascending order
  * potentially dynamically choosing an appropriate algorithm given the type and size of the array.
  * No assurance is made of the stability of the sort.
  *
  * @param a the array to be sorted.
  * @since 8.3.0
  */
 public static KEY_GENERIC void unstableSort(final KEY_GENERIC_TYPE a[]) {
  unstableSort(a, 0, a.length);
 }

 /**
 Sorts the specified range of elements according to the order induced by the specified comparator,
  * potentially dynamically choosing an appropriate algorithm given the type and size of the array.
  * No assurance is made of the stability of the sort.
  *
  * @param a the array to be sorted.
  * @param from the index of the first element (inclusive) to be sorted.
  * @param to the index of the last element (exclusive) to be sorted.
  * @param comp the comparator to determine the sorting order.
  * @since 8.3.0
  */
 public static KEY_GENERIC void unstableSort(final KEY_GENERIC_TYPE a[], final int from, final int to,
KEY_COMPARATOR KEY_GENERIC comp) {
  quickSort(a, from, to, comp);
 }

 /** Sorts an array according to the order induced by the specified comparator,
  * potentially dynamically choosing an appropriate algorithm given the type and size of the array.
  * No assurance is made of the stability of the sort.
  *
  * @param a the array to be sorted.
  * @param comp the comparator
 to determine the sorting order.
  * @since 8.3.0
  */
 public static KEY_GENERIC void unstableSort(final KEY_GENERIC_TYPE a[], KEY_COMPARATOR
KEY_GENERIC comp) {
  unstableSort(a, 0, a.length, comp);
 }
```

```
 /** Sorts the specified range of elements according to the natural ascending order using mergesort, using a given
pre-filled support array.
  *
  * <p>This sort is guaranteed to be <i>stable</i>: equal elements will not be reordered as a result
  * of the sort. Moreover, no support arrays will be allocated.
  *
  * @param a the array to be sorted.
  * @param from the index of the first element (inclusive) to be sorted.
  * @param to the index of the last element (exclusive) to be sorted.
  * @param supp a support array containing at least {@code to} elements, and whose entries are identical to those
  * of {@code a} in the specified range.
  */

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public static KEY_GENERIC void mergeSort(final KEY_GENERIC_TYPE a[], final int from, final int to,
 final KEY_GENERIC_TYPE supp[]) {
  int len = to - from;

  // Insertion sort on smallest arrays
  if (len < MERGESORT_NO_REC) {
   insertionSort(a, from, to);
   return;
  }

  // Recursively sort halves of a into supp
  final int mid = (from + to) >>> 1;
  mergeSort(supp, from, mid, a);
  mergeSort(supp, mid, to, a);

  // If list is already sorted, just copy from supp to a.  This is an
  // optimization that results in faster sorts for nearly ordered lists.
  if (KEY_LESSEQ(supp[mid - 1], supp[mid])) {
   System.arraycopy(supp, from, a, from, len);
   return;
  }

  // Merge sorted halves (now in supp) into a
  for(int i = from, p = from, q = mid; i < to; i++) {
   if (q >= to || p < mid && KEY_LESSEQ(supp[p], supp[q])) a[i] = supp[p++];
   else a[i] = supp[q++];
  }
 }

 /** Sorts the specified range of elements according to the natural ascending order using mergesort.
  *
  * <p>This sort is guaranteed to be <i>stable</i>: equal elements will not be reordered as a result
  * of the
```

```
sort. An array as large as {@code a} will be allocated by this method.

 * @param a the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
public static KEY_GENERIC void mergeSort(final KEY_GENERIC_TYPE a[], final int from, final int to) {
 mergeSort(a, from, to, a.clone());
}

/** Sorts an array according to the natural ascending order using mergesort.
 *
 * <p>This sort is guaranteed to be <i>stable</i>: equal elements will not be reordered as a result
 * of the sort. An array as large as {@code a} will be allocated by this method.

 * @param a the array to be sorted.
 */
public static KEY_GENERIC void mergeSort(final KEY_GENERIC_TYPE a[]) {
 mergeSort(a, 0, a.length);
}

/** Sorts the specified range of elements according to the order induced by the specified
 * comparator using mergesort, using a given pre-filled support array.
 *
 * <p>This sort is
guaranteed to be <i>stable</i>: equal elements will not be reordered as a result
 * of the sort. Moreover, no support arrays will be allocated.

 * @param a the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 * @param comp the comparator to determine the sorting order.
 * @param supp a support array containing at least {@code to} elements, and whose entries are identical to those
 * of {@code a} in the specified range.
 */
public static KEY_GENERIC void mergeSort(final KEY_GENERIC_TYPE a[], final int from, final int to,
KEY_COMPARATOR KEY_GENERIC comp, final KEY_GENERIC_TYPE supp[]) {
 int len = to - from;

 // Insertion sort on smallest arrays
 if (len < MERGESORT_NO_REC) {
  insertionSort(a, from, to, comp);
  return;
 }

 // Recursively sort halves of a into supp
 final int mid = (from + to) >>> 1;
 mergeSort(supp, from, mid, comp, a);
```

```
 mergeSort(supp,
mid, to, comp, a);

 // If list is already sorted, just copy from supp to a.  This is an
 // optimization that results in faster sorts for nearly ordered lists.
 if (comp.compare(supp[mid - 1], supp[mid]) <= 0) {
  System.arraycopy(supp, from, a, from, len);
  return;
 }

 // Merge sorted halves (now in supp) into a
 for(int i = from, p = from, q = mid; i < to; i++) {
  if (q >= to || p < mid && comp.compare(supp[p], supp[q]) <= 0) a[i] = supp[p++];
  else a[i] = supp[q++];
 }
}

/** Sorts the specified range of elements according to the order induced by the specified
 * comparator using mergesort.
 *
 * <p>This sort is guaranteed to be <i>stable</i>: equal elements will not be reordered as a result
 * of the sort. An array as large as {@code a} will be allocated by this method.
 *
 * @param a the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 * @param
comp the comparator to determine the sorting order.
 */
 public static KEY_GENERIC void mergeSort(final KEY_GENERIC_TYPE a[], final int from, final int to,
KEY_COMPARATOR KEY_GENERIC comp) {
  mergeSort(a, from, to, comp, a.clone());
 }

/** Sorts an array according to the order induced by the specified
 * comparator using mergesort.
 *
 * <p>This sort is guaranteed to be <i>stable</i>: equal elements will not be reordered as a result
 * of the sort.  An array as large as {@code a} will be allocated by this method.

 * @param a the array to be sorted.
 * @param comp the comparator to determine the sorting order.
 */
 public static KEY_GENERIC void mergeSort(final KEY_GENERIC_TYPE a[], KEY_COMPARATOR
KEY_GENERIC comp) {
  mergeSort(a, 0, a.length, comp);
 }
```

```
/** Sorts an array according to the natural ascending order,
 * potentially dynamically choosing an appropriate algorithm given the type and size of the array. The
 * sort will be stable unless it is provable that it
would be impossible for there to be any difference
 * between a stable and unstable sort for the given type, in which case stability is meaningless and thus
 * unspecified.
 *
 * <p>An array as large as {@code a} may be allocated by this method.
 *
 * @param a the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 * @since 8.3.0
 */
public static KEY_GENERIC void stableSort(final KEY_GENERIC_TYPE a[], final int from, final int to) {
#if KEYS_REFERENCE
 // Use JDK's sort, which is likely to be adaptive and still be stable.
 java.util.Arrays.sort(a, from, to);
#elif KEYS_PRIMITIVE && !(KEY_CLASS_Float || KEY_CLASS_Double)
 // For non-floating point primitive types, when comparing naturally,
 // it is impossible to tell the difference between a stable and not-stable sort.
 // So just use the probably faster unstable sort.
 unstableSort(a, from, to);
#else
 //
Due to subtle differences between Float/Double.compare and operator compare, it is
 // not safe to delegate this to java.util.Arrays.sort(double[], int, int)
 mergeSort(a, from, to);
#endif
}

/** Sorts the specified range of elements according to the natural ascending order
 * potentially dynamically choosing an appropriate algorithm given the type and size of the array. The
 * sort will be stable unless it is provable that it would be impossible for there to be any difference
 * between a stable and unstable sort for the given type, in which case stability is meaningless and thus
 * unspecified.
 *
 * <p>An array as large as {@code a} may be allocated by this method.
 *
 * @param a the array to be sorted.
 * @since 8.3.0
 */
public static KEY_GENERIC void stableSort(final KEY_GENERIC_TYPE a[]) {
 stableSort(a, 0, a.length);
}

/** Sorts the specified range of elements according to the order induced by the specified comparator,
 * potentially dynamically choosing an
```

```
 appropriate algorithm given the type and size of the array. The
  * sort will be stable unless it is provable that it would be impossible for there to be any difference
  * between a stable and unstable sort for the given type, in which case stability is meaningless and thus
  * unspecified.
  *
  * <p>An array as large as {@code a} may be allocated by this method.
  *
  * @param a the array to be sorted.
  * @param from the index of the first element (inclusive) to be sorted.
  * @param to the index of the last element (exclusive) to be sorted.
  * @param comp the comparator to determine the sorting order.
  * @since 8.3.0
  */
 public static KEY_GENERIC void stableSort(final KEY_GENERIC_TYPE a[], final int from, final int to,
KEY_COMPARATOR KEY_GENERIC comp) {
#if KEYS_REFERENCE
  // Use JDK's sort, which is likely to be adaptive and still be stable.
  java.util.Arrays.sort(a, from, to, comp);
#else
  mergeSort(a, from, to, comp);
#endif
 }

 /** Sorts an array according to the order
 induced by the specified comparator,
  * potentially dynamically choosing an appropriate algorithm given the type and size of the array. The
  * sort will be stable unless it is provable that it would be impossible for there to be any difference
  * between a stable and unstable sort for the given type, in which case stability is meaningless and thus
  * unspecified.
  *
  * <p>An array as large as {@code a} may be allocated by this method.
  *
  * @param a the array to be sorted.
  * @param comp the comparator to determine the sorting order.
  * @since 8.3.0
  */
 public static KEY_GENERIC void stableSort(final KEY_GENERIC_TYPE a[], KEY_COMPARATOR
KEY_GENERIC comp) {
  stableSort(a, 0, a.length, comp);
 }

#if ! KEY_CLASS_Boolean

 /**
  * Searches a range of the specified array for the specified value using
  * the binary search algorithm. The range must be sorted prior to making this call.
  * If it is not sorted, the results are undefined. If the range contains multiple elements
```

```
with
 * the specified value, there is no guarantee which one will be found.
 *
 * @param a the array to be searched.
 * @param from  the index of the first element (inclusive) to be searched.
 * @param to  the index of the last element (exclusive) to be searched.
 * @param key the value to be searched for.
 * @return index of the search key, if it is contained in the array;
 *          otherwise, {@code (-(<i>insertion point</i>) - 1)}.  The <i>insertion
 *          point</i> is defined as the the point at which the value would
 *          be inserted into the array: the index of the first
 *          element greater than the key, or the length of the array, if all
 *          elements in the array are less than the specified key.  Note
 *          that this guarantees that the return value will be &ge; 0 if
 *          and only if the key is found.
 * @see java.util.Arrays
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC int binarySearch(final
KEY_GENERIC_TYPE[] a, int from, int to, final KEY_GENERIC_TYPE key) {
 KEY_GENERIC_TYPE midVal;
 to--;
 while (from <= to) {
  final int mid = (from + to) >>> 1;
  midVal = a[mid];
#if KEYS_PRIMITIVE
  if (midVal < key) from = mid + 1;
  else if (midVal > key) to = mid - 1;
  else return mid;
#else
  final int cmp = ((Comparable KEY_SUPER_GENERIC)midVal).compareTo(key);
  if (cmp < 0) from = mid + 1;
  else if (cmp > 0) to = mid - 1;
  else return mid;
#endif
 }
 return -(from + 1);
}

/**
 * Searches an array for the specified value using
 * the binary search algorithm. The range must be sorted prior to making this call.
 * If it is not sorted, the results are undefined. If the range contains multiple elements with
 * the specified value, there is no guarantee which one will be found.
 *
 * @param a the array to be searched.
 * @param key the value to be searched for.
 * @return index of the search key, if it is contained in the array;
```

```
 *
  otherwise, {@code (-(<i>insertion point</i>) - 1)}.  The <i>insertion
 *          point</i> is defined as the the point at which the value would
 *          be inserted into the array: the index of the first
 *          element greater than the key, or the length of the array, if all
 *          elements in the array are less than the specified key.  Note
 *          that this guarantees that the return value will be &ge; 0 if
 *          and only if the key is found.
 * @see java.util.Arrays
 */
 public static KEY_GENERIC int binarySearch(final KEY_GENERIC_TYPE[] a, final KEY_GENERIC_TYPE
key) {
  return binarySearch(a, 0, a.length, key);
 }

 /**
  * Searches a range of the specified array for the specified value using
  * the binary search algorithm and a specified comparator. The range must be sorted following the comparator prior
to making this call.
  * If it is not sorted, the results are undefined. If the range contains multiple elements with
  * the
 specified value, there is no guarantee which one will be found.
 *
 * @param a the array to be searched.
 * @param from  the index of the first element (inclusive) to be searched.
 * @param to  the index of the last element (exclusive) to be searched.
 * @param key the value to be searched for.
 * @param c a comparator.
 * @return index of the search key, if it is contained in the array;
 *          otherwise, {@code (-(<i>insertion point</i>) - 1)}.  The <i>insertion
 *          point</i> is defined as the the point at which the value would
 *          be inserted into the array: the index of the first
 *          element greater than the key, or the length of the array, if all
 *          elements in the array are less than the specified key.  Note
 *          that this guarantees that the return value will be &ge; 0 if
 *          and only if the key is found.
 * @see java.util.Arrays
 */
 public static KEY_GENERIC int binarySearch(final KEY_GENERIC_TYPE[]
 a, int from, int to, final KEY_GENERIC_TYPE key, final KEY_COMPARATOR KEY_GENERIC c) {
  KEY_GENERIC_TYPE midVal;
  to--;
  while (from <= to) {
   final int mid = (from + to) >>> 1;
   midVal = a[mid];
   final int cmp = c.compare(midVal, key);
   if (cmp < 0) from = mid + 1;
   else if (cmp > 0) to = mid - 1;
```

```
  else return mid; // key found
  }
  return -(from + 1);
 }

 /**
  * Searches an array for the specified value using
  * the binary search algorithm and a specified comparator. The range must be sorted following the comparator prior
to making this call.
  * If it is not sorted, the results are undefined. If the range contains multiple elements with
  * the specified value, there is no guarantee which one will be found.
  *
  * @param a the array to be searched.
  * @param key the value to be searched for.
  * @param c a comparator.
  * @return index of the search key, if it is contained in the array;
  *          otherwise, {@code (-(<i>insertion point</i>) - 1)}.
  The <i>insertion
  *          point</i> is defined as the the point at which the value would
  *          be inserted into the array: the index of the first
  *          element greater than the key, or the length of the array, if all
  *          elements in the array are less than the specified key.  Note
  *          that this guarantees that the return value will be &ge; 0 if
  *          and only if the key is found.
  * @see java.util.Arrays
  */
 public static KEY_GENERIC int binarySearch(final KEY_GENERIC_TYPE[] a, final KEY_GENERIC_TYPE
key, final KEY_COMPARATOR KEY_GENERIC c) {
  return binarySearch(a, 0, a.length, key, c);
 }


#if KEYS_PRIMITIVE
 /** The size of a digit used during radix sort (must be a power of 2). */
 private static final int DIGIT_BITS = 8;
 /** The mask to extract a digit of {@link #DIGIT_BITS} bits. */
 private static final int DIGIT_MASK = (1 << DIGIT_BITS) - 1;
 /** The number of digits per element. */
 private static final int
 DIGITS_PER_ELEMENT = KEY_CLASS.SIZE / DIGIT_BITS;
 private static final int RADIXSORT_NO_REC = 1024;
 private static final int PARALLEL_RADIXSORT_NO_FORK = 1024;

 // The thresholds were determined on an Intel i7 8700K.
 /** Threshold <em>hint</em> for using a radix sort vs a comparison based sort. */
#if KEY_CLASS_Byte || KEY_CLASS_Short
 static final int RADIX_SORT_MIN_THRESHOLD = 1000;
#elif KEY_CLASS_Long || KEY_CLASS_Double
```

```
 static final int RADIX_SORT_MIN_THRESHOLD = 4000;
#else
 static final int RADIX_SORT_MIN_THRESHOLD = 2000;
#endif

 /** This method fixes negative numbers so that the combination exponent/significand is lexicographically sorted. */
#if KEY_CLASS_Double
 private static final long fixDouble(final double d) {
  final long l = Double.doubleToLongBits(d);
  return l >= 0 ? l : l ^ 0x7FFFFFFFFFFFFFFFL;
 }
#elif KEY_CLASS_Float
 private static final int fixFloat(final float f) {
  final int i = Float.floatToIntBits(f);
  return i >= 0 ? i : i ^ 0x7FFFFFFF;
 }
#endif


 /**
 Sorts the specified array using radix sort.
  *
  * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
  * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
  *
  * <p>This implementation is significantly faster than quicksort
  * already at small sizes (say, more than 5000 elements), but it can only
  * sort in ascending order.
  *
  * @param a the array to be sorted.
  */
 public static void radixSort(final KEY_TYPE[] a) {
  radixSort(a, 0, a.length);
 }

 /** Sorts the specified range of an array using radix sort.
  *
  * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
  * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
  *
  * <p>This implementation is significantly faster than quicksort
  * already at small sizes (say, more than 5000 elements), but it can
 only
  * sort in ascending order.
  *
  * @param a the array to be sorted.
  * @param from the index of the first element (inclusive) to be sorted.
  * @param to the index of the last element (exclusive) to be sorted.
```

```
 */
 public static void radixSort(final KEY_TYPE[] a, final int from, final int to) {
  if (to - from < RADIXSORT_NO_REC) {
   quickSort(a, from, to);
   return;
  }

  final int maxLevel = DIGITS_PER_ELEMENT - 1;

  final int stackSize = ((1 << DIGIT_BITS) - 1) * (DIGITS_PER_ELEMENT - 1) + 1;
  int stackPos = 0;
  final int[] offsetStack = new int[stackSize];
  final int[] lengthStack = new int[stackSize];
  final int[] levelStack = new int[stackSize];

  offsetStack[stackPos] = from;
  lengthStack[stackPos] = to - from;
  levelStack[stackPos++] = 0;

  final int[] count = new int[1 << DIGIT_BITS];
  final int[] pos = new int[1 << DIGIT_BITS];

  while(stackPos > 0) {
   final int first = offsetStack[--stackPos];
   final int length = lengthStack[stackPos];
   final int
 level = levelStack[stackPos];
#if KEY_CLASS_Character
   final int signMask = 0;
#else
   final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
#endif
   final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
shift that extract the right byte from a key

   // Count keys.
   for(int i = first + length; i-- != first;) count[INT(KEY2LEXINT(a[i]) >>> shift & DIGIT_MASK ^ signMask)]++;
   // Compute cumulative distribution
   int lastUsed = -1;
   for (int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
    if (count[i] != 0) lastUsed = i;
    pos[i] = (p += count[i]);
   }

   final int end = first + length - count[lastUsed];
   // i moves through the start of each block
   for(int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
    KEY_TYPE t = a[i];
```

```
   c = INT(KEY2LEXINT(t) >>> shift & DIGIT_MASK ^ signMask);

   if (i < end) { // When all slots are OK, the last slot is necessarily OK.
    while ((d
= --pos[c]) > i) {
     final KEY_TYPE z = t;
     t = a[d];
     a[d] = z;
     c = INT(KEY2LEXINT(t) >>> shift & DIGIT_MASK ^ signMask);
    }
    a[i] = t;
   }

   if (level < maxLevel && count[c] > 1) {
    if (count[c] < RADIXSORT_NO_REC) quickSort(a, i, i + count[c]);
    else {
     offsetStack[stackPos] = i;
     lengthStack[stackPos] = count[c];
     levelStack[stackPos++] = level + 1;
    }
   }
  }
 }
}

protected static final class Segment {
 protected final int offset, length, level;
 protected Segment(final int offset, final int length, final int level) {
  this.offset = offset;
  this.length = length;
  this.level = level;
 }

 @Override
 public String toString() { return "Segment [offset=" + offset + ", length=" + length + ", level=" + level + "]"; }
}

protected static final Segment POISON_PILL = new Segment(-1, -1, -1);


/** Sorts the specified range of an array using parallel radix sort.
 *
 * <p>The sorting algorithm is a tuned radix
sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This implementation uses a pool of {@link Runtime#availableProcessors()} threads.
 *
```

```
 * @param a the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
 public static void parallelRadixSort(final KEY_TYPE[] a, final int from, final int to) {
  if (to - from < PARALLEL_RADIXSORT_NO_FORK) {
   quickSort(a, from, to);
   return;
  }
  final int maxLevel = DIGITS_PER_ELEMENT - 1;
  final LinkedBlockingQueue<Segment> queue = new LinkedBlockingQueue<>();
  queue.add(new Segment(from, to - from, 0));
  final AtomicInteger queueSize = new AtomicInteger(1);
  final int numberOfThreads = Runtime.getRuntime().availableProcessors();
  final ExecutorService executorService = Executors.newFixedThreadPool(numberOfThreads,
 Executors.defaultThreadFactory());
  final ExecutorCompletionService<Void> executorCompletionService = new
ExecutorCompletionService<>(executorService);

  for(int j = numberOfThreads; j-- != 0;) executorCompletionService.submit(() -> {
   final int[] count = new int[1 << DIGIT_BITS];
   final int[] pos = new int[1 << DIGIT_BITS];

   for(;;) {
    if (queueSize.get() == 0) for(int i = numberOfThreads; i-- != 0;) queue.add(POISON_PILL);
    final Segment segment = queue.take();
    if (segment == POISON_PILL) return null;

    final int first = segment.offset;
    final int length = segment.length;
    final int level = segment.level;

#if KEY_CLASS_Character
    final int signMask = 0;
#else
    final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
#endif
    final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
shift that extract the right byte from a key

    // Count keys.
    for(int i
 = first + length; i-- != first;) count[INT(KEY2LEXINT(a[i]) >>> shift & DIGIT_MASK ^ signMask)]++;
    // Compute cumulative distribution
    int lastUsed = -1;
    for(int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
     if (count[i] != 0) lastUsed = i;
     pos[i] = (p += count[i]);
```

```
    }

    final int end = first + length - count[lastUsed];
    // i moves through the start of each block
    for(int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
     KEY_TYPE t = a[i];
     c = INT(KEY2LEXINT(t) >>> shift & DIGIT_MASK ^ signMask);
     if (i < end) {
      while((d = --pos[c]) > i) {
       final KEY_TYPE z = t;
       t = a[d];
       a[d] = z;
       c = INT(KEY2LEXINT(t) >>> shift & DIGIT_MASK ^ signMask);
      }
      a[i] = t;
     }

     if (level < maxLevel && count[c] > 1) {
      if (count[c] < PARALLEL_RADIXSORT_NO_FORK) quickSort(a, i, i + count[c]);
      else {
       queueSize.incrementAndGet();
       queue.add(new Segment(i, count[c], level + 1));
      }
     }
    }
    queueSize.decrementAndGet();
   }
  });

  Throwable
 problem = null;
  for(int i = numberOfThreads; i-- != 0;)
   try {
    executorCompletionService.take().get();
   }
   catch(Exception e) {
    problem = e.getCause(); // We keep only the last one. They will be logged anyway.
   }

  executorService.shutdown();
  if (problem != null) throw (problem instanceof RuntimeException) ? (RuntimeException)problem : new
RuntimeException(problem);
  }

 /** Sorts the specified array using parallel radix sort.
  *
  * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
  * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
```

```
 *
 * <p>This implementation uses a pool of {@link Runtime#availableProcessors()} threads.
 *
 * @param a the array to be sorted.
 */
 public static void parallelRadixSort(final KEY_TYPE[] a) {
  parallelRadixSort(a, 0, a.length);
 }

 /** Sorts
 the specified array using indirect radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implement an <em>indirect</em> sort. The elements of {@code perm} (which must
 * be exactly the numbers in the interval {@code [0..perm.length)}) will be permuted so that
 * {@code a[perm[i]] &le; a[perm[i + 1]]}.
 *
 * <p>This implementation will allocate, in the stable case, a support array as large as {@code perm} (note that the
stable
 * version is slightly faster).
 *
 * @param perm a permutation array indexing {@code a}.
 * @param a the array to be sorted.
 * @param stable whether the sorting algorithm should be stable.
 */
 public static void radixSortIndirect(final int[] perm, final KEY_TYPE[] a, final boolean stable) {
  radixSortIndirect(perm, a, 0, perm.length, stable);
 }

 /**
 Sorts the specified array using indirect radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implement an <em>indirect</em> sort. The elements of {@code perm} (which must
 * be exactly the numbers in the interval {@code [0..perm.length)}) will be permuted so that
 * {@code a[perm[i]] &le; a[perm[i + 1]]}.
 *
 * <p>This implementation will allocate, in the stable case, a support array as large as {@code perm} (note that the
stable
 * version is slightly faster).
 *
 * @param perm a permutation array indexing {@code a}.
 * @param a the array to be sorted.
 * @param from the index of the first element of {@code perm} (inclusive) to be permuted.
```

```
  * @param to the index of the last element of {@code perm} (exclusive) to be permuted.
  * @param stable whether the sorting algorithm
 should be stable.
  */
 public static void radixSortIndirect(final int[] perm, final KEY_TYPE[] a, final int from, final int to, final boolean
stable) {
  if (to - from < RADIXSORT_NO_REC) {
   insertionSortIndirect(perm, a, from, to);
   return;
  }

  final int maxLevel = DIGITS_PER_ELEMENT - 1;

  final int stackSize = ((1 << DIGIT_BITS) - 1) * (DIGITS_PER_ELEMENT - 1) + 1;
  int stackPos = 0;
  final int[] offsetStack = new int[stackSize];
  final int[] lengthStack = new int[stackSize];
  final int[] levelStack = new int[stackSize];

  offsetStack[stackPos] = from;
  lengthStack[stackPos] = to - from;
  levelStack[stackPos++] = 0;

  final int[] count = new int[1 << DIGIT_BITS];
  final int[] pos = new int[1 << DIGIT_BITS];
  final int[] support = stable ? new int[perm.length] : null;

  while(stackPos > 0) {
   final int first = offsetStack[--stackPos];
   final int length = lengthStack[stackPos];
   final int level = levelStack[stackPos];
#if KEY_CLASS_Character
   final int
 signMask = 0;
#else
   final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
#endif

   final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
shift that extract the right byte from a key

   // Count keys.
   for(int i = first + length; i-- != first;) count[INT(KEY2LEXINT(a[perm[i]]) >>> shift & DIGIT_MASK ^
signMask)]++;
   // Compute cumulative distribution
   int lastUsed = -1;
   for (int i = 0, p = stable ? 0 : first; i < 1 << DIGIT_BITS; i++) {
    if (count[i] != 0) lastUsed = i;
```

```
    pos[i] = (p += count[i]);
   }

   if (stable) {
    for(int i = first + length; i-- != first;) support[--pos[INT(KEY2LEXINT(a[perm[i]]) >>> shift & DIGIT_MASK ^
signMask)]] = perm[i];
    System.arraycopy(support, 0, perm, first, length);
    for(int i = 0, p = first; i <= lastUsed; i++) {
     if (level < maxLevel && count[i] > 1) {
      if (count[i] < RADIXSORT_NO_REC) insertionSortIndirect(perm, a, p, p + count[i]);
      else
 {
       offsetStack[stackPos] = p;
       lengthStack[stackPos] = count[i];
       levelStack[stackPos++] = level + 1;
      }
     }
     p += count[i];
    }
    java.util.Arrays.fill(count, 0);
   }
   else {
    final int end = first + length - count[lastUsed];
    // i moves through the start of each block
    for(int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
     int t = perm[i];
     c = INT(KEY2LEXINT(a[t]) >>> shift & DIGIT_MASK ^ signMask);

     if (i < end) { // When all slots are OK, the last slot is necessarily OK.
      while((d = --pos[c]) > i) {
       final int z = t;
       t = perm[d];
       perm[d] = z;
       c = INT(KEY2LEXINT(a[t]) >>> shift & DIGIT_MASK ^ signMask);
      }
      perm[i] = t;
     }

     if (level < maxLevel && count[c] > 1) {
      if (count[c] < RADIXSORT_NO_REC) insertionSortIndirect(perm, a, i, i + count[c]);
      else {
       offsetStack[stackPos] = i;
       lengthStack[stackPos] = count[c];
       levelStack[stackPos++]
 = level + 1;
      }
     }
    }
```

```
  }
 }
}

/** Sorts the specified range of an array using parallel indirect radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implement an <em>indirect</em> sort. The elements of {@code perm} (which must
 * be exactly the numbers in the interval {@code [0..perm.length)}) will be permuted so that
 * {@code a[perm[i]] &le; a[perm[i + 1]]}.
 *
 * <p>This implementation uses a pool of {@link Runtime#availableProcessors()} threads.
 *
 * @param perm a permutation array indexing {@code a}.
 * @param a the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 * @param stable whether the sorting algorithm should be stable.

 */
 public static void parallelRadixSortIndirect(final int perm[], final KEY_TYPE[] a, final int from, final int to, final
boolean stable) {
  if (to - from < PARALLEL_RADIXSORT_NO_FORK) {
   radixSortIndirect(perm, a, from, to, stable);
   return;
  }
  final int maxLevel = DIGITS_PER_ELEMENT - 1;
  final LinkedBlockingQueue<Segment> queue = new LinkedBlockingQueue<>();
  queue.add(new Segment(from, to - from, 0));
  final AtomicInteger queueSize = new AtomicInteger(1);
  final int numberOfThreads = Runtime.getRuntime().availableProcessors();
  final ExecutorService executorService = Executors.newFixedThreadPool(numberOfThreads,
Executors.defaultThreadFactory());
  final ExecutorCompletionService<Void> executorCompletionService = new
ExecutorCompletionService<>(executorService);
  final int[] support = stable ? new int[perm.length] : null;

  for(int j = numberOfThreads; j-- != 0;) executorCompletionService.submit(() -> {
   final int[] count = new int[1 << DIGIT_BITS];
   final int[]
 pos = new int[1 << DIGIT_BITS];

   for(;;) {
    if (queueSize.get() == 0) for(int i = numberOfThreads; i-- != 0;) queue.add(POISON_PILL);
    final Segment segment = queue.take();
    if (segment == POISON_PILL) return null;
```

```
    final int first = segment.offset;
    final int length = segment.length;
    final int level = segment.level;

#if KEY_CLASS_Character
    final int signMask = 0;
#else
    final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
#endif
    final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
shift that extract the right byte from a key



    // Count keys.
    for(int i = first + length; i-- != first;) count[INT(KEY2LEXINT(a[perm[i]]) >>> shift & DIGIT_MASK ^
signMask)]++;
    // Compute cumulative distribution
    int lastUsed = -1;
    for (int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
     if (count[i] != 0) lastUsed = i;
     pos[i] = (p += count[i]);
    }

    if (stable)
 {
     for(int i = first + length; i-- != first;) support[--pos[INT(KEY2LEXINT(a[perm[i]]) >>> shift & DIGIT_MASK ^
signMask)]] = perm[i];
     System.arraycopy(support, first, perm, first, length);
     for(int i = 0, p = first; i <= lastUsed; i++) {
      if (level < maxLevel && count[i] > 1) {
       if (count[i] < PARALLEL_RADIXSORT_NO_FORK) radixSortIndirect(perm, a, p, p + count[i], stable);
       else {
        queueSize.incrementAndGet();
        queue.add(new Segment(p, count[i], level + 1));
       }
      }
      p += count[i];
     }
     java.util.Arrays.fill(count, 0);
    }
    else {
     final int end = first + length - count[lastUsed];
     // i moves through the start of each block
     for(int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
      int t = perm[i];
      c = INT(KEY2LEXINT(a[t]) >>> shift & DIGIT_MASK ^ signMask);
```

```
     if (i < end) { // When all slots are OK, the last slot is necessarily OK.
      while((d = --pos[c]) > i) {
       final
int z = t;
       t = perm[d];
       perm[d] = z;
       c = INT(KEY2LEXINT(a[t]) >>> shift & DIGIT_MASK ^ signMask);
      }
      perm[i] = t;
     }

     if (level < maxLevel && count[c] > 1) {
      if (count[c] < PARALLEL_RADIXSORT_NO_FORK) radixSortIndirect(perm, a, i, i + count[c], stable);
      else {
       queueSize.incrementAndGet();
       queue.add(new Segment(i, count[c], level + 1));
      }
     }
    }
   }
   queueSize.decrementAndGet();
  }
 });

 Throwable problem = null;
 for(int i = numberOfThreads; i-- != 0;)
  try {
   executorCompletionService.take().get();
  }
  catch(Exception e) {
   problem = e.getCause(); // We keep only the last one. They will be logged anyway.
  }

 executorService.shutdown();
 if (problem != null) throw (problem instanceof RuntimeException) ? (RuntimeException)problem : new
RuntimeException(problem);
 }

 /** Sorts the specified array using parallel indirect radix sort.
  *
  * <p>The sorting algorithm
 is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
  * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
  *
  * <p>This method implement an <em>indirect</em> sort. The elements of {@code perm} (which must
  * be exactly the numbers in the interval {@code [0..perm.length)}) will be permuted so that
  * {@code a[perm[i]] &le; a[perm[i + 1]]}.
  *
```

```
 * <p>This implementation uses a pool of {@link Runtime#availableProcessors()} threads.
 *
 * @param perm a permutation array indexing {@code a}.
 * @param a the array to be sorted.
 * @param stable whether the sorting algorithm should be stable.
 */
public static void parallelRadixSortIndirect(final int perm[], final KEY_TYPE[] a, final boolean stable) {
 parallelRadixSortIndirect(perm, a, 0, a.length, stable);
}

/** Sorts the specified pair of arrays lexicographically using radix sort.
 * <p>The sorting algorithm is a tuned radix sort adapted from
Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of elements
 * in the same position in the two provided arrays will be considered a single key, and permuted
 * accordingly. In the end, either {@code a[i] &lt; a[i + 1]} or {@code a[i] == a[i + 1]} and {@code b[i] &le; b[i +
1]}.
 *
 * @param a the first array to be sorted.
 * @param b the second array to be sorted.
 */

public static void radixSort(final KEY_TYPE[] a, final KEY_TYPE[] b) {
 ensureSameLength(a, b);
 radixSort(a, b, 0, a.length);
}

/** Sorts the specified range of elements of two arrays using radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27
(1993).
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of elements
 * in the same position in the two provided arrays will be considered a single key, and permuted
 * accordingly. In the end, either {@code a[i] &lt; a[i + 1]} or {@code a[i] == a[i + 1]} and {@code b[i] &le; b[i +
1]}.
 *
 * @param a the first array to be sorted.
 * @param b the second array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
public static void radixSort(final KEY_TYPE[] a, final KEY_TYPE[] b, final int from, final int to) {
 if (to - from < RADIXSORT_NO_REC) {
  selectionSort(a, b, from, to);
```

```
  return;
 }

 final int layers = 2;
 final int maxLevel = DIGITS_PER_ELEMENT * layers - 1;

 final int stackSize = ((1 << DIGIT_BITS) - 1) * (layers * DIGITS_PER_ELEMENT - 1) + 1;
 int stackPos = 0;
 final int[] offsetStack = new int[stackSize];
 final
int[] lengthStack = new int[stackSize];
 final int[] levelStack = new int[stackSize];

 offsetStack[stackPos] = from;
 lengthStack[stackPos] = to - from;
 levelStack[stackPos++] = 0;

 final int[] count = new int[1 << DIGIT_BITS];
 final int[] pos = new int[1 << DIGIT_BITS];

 while(stackPos > 0) {
  final int first = offsetStack[--stackPos];
  final int length = lengthStack[stackPos];
  final int level = levelStack[stackPos];
#if KEY_CLASS_Character
  final int signMask = 0;
#else
  final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
#endif

  final KEY_TYPE[] k = level < DIGITS_PER_ELEMENT ? a : b; // This is the key array
  final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
shift that extract the right byte from a key

  // Count keys.
  for(int i = first + length; i-- != first;) count[INT(KEY2LEXINT(k[i]) >>> shift & DIGIT_MASK ^ signMask)]++;

  // Compute cumulative distribution
  int
lastUsed = -1;
  for (int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
   if (count[i] != 0) lastUsed = i;
   pos[i] = (p += count[i]);
  }


  final int end = first + length - count[lastUsed];
  // i moves through the start of each block
```

```
  for(int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
   KEY_TYPE t = a[i];
   KEY_TYPE u = b[i];
   c = INT(KEY2LEXINT(k[i]) >>> shift & DIGIT_MASK ^ signMask);

   if (i < end) { // When all slots are OK, the last slot is necessarily OK.
    while((d = --pos[c]) > i) {
     c = INT(KEY2LEXINT(k[d]) >>> shift & DIGIT_MASK ^ signMask);
     KEY_TYPE z = t;
     t = a[d];
     a[d] = z;
     z = u;
     u = b[d];
     b[d] = z;
    }
    a[i] = t;
    b[i] = u;

   }

   if (level < maxLevel && count[c] > 1) {
    if (count[c] < RADIXSORT_NO_REC) selectionSort(a, b, i, i + count[c]);
    else {
     offsetStack[stackPos] = i;
     lengthStack[stackPos] = count[c];
     levelStack[stackPos++]
 = level + 1;
    }
   }
  }
 }
}

/** Sorts the specified range of elements of two arrays using a parallel radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of elements
 * in the same position in the two provided arrays will be considered a single key, and permuted
 * accordingly. In the end, either {@code a[i] &lt; a[i + 1]} or {@code a[i] == a[i + 1]} and {@code b[i] &le; b[i +
1]}.
 *
 * <p>This implementation uses a pool of {@link Runtime#availableProcessors()} threads.
 *
 * @param a the first array to be sorted.
 * @param b the second array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
```

```
 * @param to the index of the last element (exclusive) to be sorted.

*/
public static void parallelRadixSort(final KEY_TYPE[] a, final KEY_TYPE[] b, final int from, final int to) {
 if (to - from < PARALLEL_RADIXSORT_NO_FORK) {
  quickSort(a, b, from, to);
  return;
 }
 final int layers = 2;
 if (a.length != b.length) throw new IllegalArgumentException("Array size mismatch.");
 final int maxLevel = DIGITS_PER_ELEMENT * layers - 1;
 final LinkedBlockingQueue<Segment> queue = new LinkedBlockingQueue<>();
 queue.add(new Segment(from, to - from, 0));
 final AtomicInteger queueSize = new AtomicInteger(1);
 final int numberOfThreads = Runtime.getRuntime().availableProcessors();
 final ExecutorService executorService = Executors.newFixedThreadPool(numberOfThreads,
Executors.defaultThreadFactory());
 final ExecutorCompletionService<Void> executorCompletionService = new
ExecutorCompletionService<>(executorService);
 for (int j = numberOfThreads; j-- != 0;) executorCompletionService.submit(() -> {
  final int[] count = new int[1 << DIGIT_BITS];
  final
int[] pos = new int[1 << DIGIT_BITS];
  for (;;) {
   if (queueSize.get() == 0) for (int i = numberOfThreads; i-- != 0;)
    queue.add(POISON_PILL);
   final Segment segment = queue.take();
   if (segment == POISON_PILL) return null;
   final int first = segment.offset;
   final int length = segment.length;
   final int level = segment.level;
   final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
   final KEY_TYPE[] k = level < DIGITS_PER_ELEMENT ? a : b; // This is the key array
   final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS;
   // Count keys.
   for (int i = first + length; i-- != first;)
    count[INT(KEY2LEXINT(k[i]) >>> shift & DIGIT_MASK ^ signMask)]++;
   // Compute cumulative distribution
   int lastUsed = -1;
   for (int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
    if (count[i] != 0) lastUsed = i;
    pos[i] = (p += count[i]);
   }
   final int end =
first + length - count[lastUsed];
   for (int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
    KEY_TYPE t = a[i];
    KEY_TYPE u = b[i];
```

```
      c = INT(KEY2LEXINT(k[i]) >>> shift & DIGIT_MASK ^ signMask);
      if (i < end) { // When all slots are OK, the last slot is necessarily OK.
       while ((d = --pos[c]) > i) {
        c = INT(KEY2LEXINT(k[d]) >>> shift & DIGIT_MASK ^ signMask);
        final KEY_TYPE z = t;
        final KEY_TYPE w = u;
        t = a[d];
        u = b[d];
        a[d] = z;
        b[d] = w;
       }
       a[i] = t;
       b[i] = u;
      }
      if (level < maxLevel && count[c] > 1) {
       if (count[c] < PARALLEL_RADIXSORT_NO_FORK) quickSort(a, b, i, i + count[c]);
       else {
        queueSize.incrementAndGet();
        queue.add(new Segment(i, count[c], level + 1));
       }
      }
     }
     queueSize.decrementAndGet();
    }
   });
  Throwable problem = null;
  for (int i = numberOfThreads; i-- != 0;)
   try {
    executorCompletionService.take().get();
   }
   catch
  (Exception e) {
    problem = e.getCause(); // We keep only the last one. They will be logged anyway.
   }
  executorService.shutdown();
  if (problem != null) throw (problem instanceof RuntimeException) ? (RuntimeException)problem : new
RuntimeException(problem);
  }



 /** Sorts two arrays using a parallel radix sort.
  *
  * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
  * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
  *
  * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of elements
  * in the same position in the two provided arrays will be considered a single key, and permuted
  * accordingly. In the end, either {@code a[i] &lt; a[i + 1]} or {@code a[i] == a[i + 1]} and {@code b[i] &le; b[i +
```

```
1]}.
 *
 * <p>This implementation uses a pool of {@link Runtime#availableProcessors()} threads.

 *
 * @param a the first array to be sorted.
 * @param b the second array to be sorted.
 */
 public static void parallelRadixSort(final KEY_TYPE[] a, final KEY_TYPE[] b) {
  ensureSameLength(a, b);
  parallelRadixSort(a, b, 0, a.length);
 }


 private static KEY_GENERIC void insertionSortIndirect(final int[] perm, final KEY_TYPE[] a, final KEY_TYPE[]
b, final int from, final int to) {
  for (int i = from; ++i < to;) {
   int t = perm[i];
   int j = i;
   for (int u = perm[j - 1]; KEY_LESS(a[t], a[u]) || KEY_CMP_EQ(a[t], a[u]) && KEY_LESS(b[t], b[u]); u = perm[--j
- 1]) {
    perm[j] = u;
    if (from == j - 1) {
     --j;
     break;
    }
   }
   perm[j] = t;
  }
 }

 /** Sorts the specified pair of arrays lexicographically using indirect radix sort.
  *
  * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
  * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
  *
  *
 <p>This method implement an <em>indirect</em> sort. The elements of {@code perm} (which must
  * be exactly the numbers in the interval {@code [0..perm.length)}) will be permuted so that
  * {@code a[perm[i]] &le; a[perm[i + 1]]} or {@code a[perm[i]] == a[perm[i + 1]]} and {@code b[perm[i]] &le;
b[perm[i + 1]]}.
  *
  * <p>This implementation will allocate, in the stable case, a further support array as large as {@code perm} (note
that the stable
  * version is slightly faster).
  *
  * @param perm a permutation array indexing {@code a}.
  * @param a the array to be sorted.
```

```
 * @param b the second array to be sorted.
 * @param stable whether the sorting algorithm should be stable.
 */
 public static void radixSortIndirect(final int[] perm, final KEY_TYPE[] a, final KEY_TYPE[] b, final boolean
stable) {
  ensureSameLength(a, b);
  radixSortIndirect(perm, a, b, 0, a.length, stable);
 }

 /** Sorts the specified pair of arrays lexicographically using indirect radix sort.
  *
  * <p>The
 sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
  * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
  *
  * <p>This method implement an <em>indirect</em> sort. The elements of {@code perm} (which must
  * be exactly the numbers in the interval {@code [0..perm.length)}) will be permuted so that
  * {@code a[perm[i]] &le; a[perm[i + 1]]} or {@code a[perm[i]] == a[perm[i + 1]]} and {@code b[perm[i]] &le;
b[perm[i + 1]]}.
  *
  * <p>This implementation will allocate, in the stable case, a further support array as large as {@code perm} (note
that the stable
  * version is slightly faster).
  *
  * @param perm a permutation array indexing {@code a}.
  * @param a the array to be sorted.
  * @param b the second array to be sorted.
  * @param from the index of the first element of {@code perm} (inclusive) to be permuted.
  * @param to the index of the last element of {@code perm} (exclusive)
 to be permuted.
  * @param stable whether the sorting algorithm should be stable.
  */
 public static void radixSortIndirect(final int[] perm, final KEY_TYPE[] a, final KEY_TYPE[] b, final int from, final
int to, final boolean stable) {
  if (to - from < RADIXSORT_NO_REC) {
   insertionSortIndirect(perm, a, b, from, to);
   return;
  }

  final int layers = 2;
  final int maxLevel = DIGITS_PER_ELEMENT * layers - 1;

  final int stackSize = ((1 << DIGIT_BITS) - 1) * (layers * DIGITS_PER_ELEMENT - 1) + 1;
  int stackPos = 0;
  final int[] offsetStack = new int[stackSize];
  final int[] lengthStack = new int[stackSize];
  final int[] levelStack = new int[stackSize];
```

```
  offsetStack[stackPos] = from;
  lengthStack[stackPos] = to - from;
  levelStack[stackPos++] = 0;

  final int[] count = new int[1 << DIGIT_BITS];
  final int[] pos = new int[1 << DIGIT_BITS];
  final int[] support = stable ? new int[perm.length] : null;

  while(stackPos > 0) {
   final int first = offsetStack[--stackPos];
   final
 int length = lengthStack[stackPos];
   final int level = levelStack[stackPos];
#if KEY_CLASS_Character
   final int signMask = 0;
#else
   final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
#endif

   final KEY_TYPE[] k = level < DIGITS_PER_ELEMENT ? a : b; // This is the key array
   final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
shift that extract the right byte from a key

   // Count keys.
   for(int i = first + length; i-- != first;) count[INT(KEY2LEXINT(k[perm[i]]) >>> shift & DIGIT_MASK ^
signMask)]++;

   // Compute cumulative distribution
   int lastUsed = -1;
   for (int i = 0, p = stable ? 0 : first; i < 1 << DIGIT_BITS; i++) {
    if (count[i] != 0) lastUsed = i;
    pos[i] = (p += count[i]);
   }

   if (stable) {
    for(int i = first + length; i-- != first;) support[--pos[INT(KEY2LEXINT(k[perm[i]]) >>> shift & DIGIT_MASK ^
signMask)]] = perm[i];
    System.arraycopy(support, 0,
 perm, first, length);
    for(int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
     if (level < maxLevel && count[i] > 1) {
      if (count[i] < RADIXSORT_NO_REC) insertionSortIndirect(perm, a, b, p, p + count[i]);
      else {
       offsetStack[stackPos] = p;
       lengthStack[stackPos] = count[i];
       levelStack[stackPos++] = level + 1;
      }
     }
```

```
    p += count[i];
   }
   java.util.Arrays.fill(count, 0);
  }
  else {
   final int end = first + length - count[lastUsed];
   // i moves through the start of each block
   for(int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
    int t = perm[i];
    c = INT(KEY2LEXINT(k[t]) >>> shift & DIGIT_MASK ^ signMask);

    if (i < end) { // When all slots are OK, the last slot is necessarily OK.
     while((d = --pos[c]) > i) {
      final int z = t;
      t = perm[d];
      perm[d] = z;
      c = INT(KEY2LEXINT(k[t]) >>> shift & DIGIT_MASK ^ signMask);
     }
     perm[i] = t;
    }



    if (level
< maxLevel && count[c] > 1) {
     if (count[c] < RADIXSORT_NO_REC) insertionSortIndirect(perm, a, b, i, i + count[c]);
     else {
      offsetStack[stackPos] = i;
      lengthStack[stackPos] = count[c];
      levelStack[stackPos++] = level + 1;
     }
    }
   }
  }
 }
}




private static void selectionSort(final KEY_TYPE[][] a, final int from, final int to, final int level) {
 final int layers = a.length;
 final int firstLayer = level / DIGITS_PER_ELEMENT;

 for(int i = from; i < to - 1; i++) {
  int m = i;
  for(int j = i + 1; j < to; j++) {
   for(int p = firstLayer; p < layers; p++) {
    if (a[p][j] < a[p][m]) {
```

```
     m = j;
     break;
    }
    else if (a[p][j] > a[p][m]) break;
   }
  }
  if (m != i) {
   for(int p = layers; p-- != 0;) {
    final KEY_TYPE u = a[p][i];
    a[p][i] = a[p][m];
    a[p][m] = u;
   }
  }
 }
}
```

```
/** Sorts the specified array of arrays lexicographically using radix sort.
 *
 * <p>The sorting algorithm is a tuned
radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the provided arrays. Tuples of elements
 * in the same position will be considered a single key, and permuted
 * accordingly.
 *
 * @param a an array containing arrays of equal length to be sorted lexicographically in parallel.
 */
public static void radixSort(final KEY_TYPE[][] a) {
 radixSort(a, 0, a[0].length);
}

/** Sorts the specified array of arrays lexicographically using radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the provided arrays. Tuples of elements
 * in the
same position will be considered a single key, and permuted
 * accordingly.
 *
 * @param a an array containing arrays of equal length to be sorted lexicographically in parallel.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
```

```
 public static void radixSort(final KEY_TYPE[][] a, final int from, final int to) {
  if (to - from < RADIXSORT_NO_REC) {
   selectionSort(a, from, to, 0);
   return;
  }

  final int layers = a.length;
  final int maxLevel = DIGITS_PER_ELEMENT * layers - 1;
  for(int p = layers, l = a[0].length; p-- != 0;) if (a[p].length != l) throw new IllegalArgumentException("The array of
index " + p + " has not the same length of the array of index 0.");

  final int stackSize = ((1 << DIGIT_BITS) - 1) * (layers * DIGITS_PER_ELEMENT - 1) + 1;
  int stackPos = 0;
  final int[] offsetStack = new int[stackSize];
  final int[] lengthStack = new int[stackSize];
  final int[]
 levelStack = new int[stackSize];

  offsetStack[stackPos] = from;
  lengthStack[stackPos] = to - from;
  levelStack[stackPos++] = 0;

  final int[] count = new int[1 << DIGIT_BITS];
  final int[] pos = new int[1 << DIGIT_BITS];
  final KEY_TYPE[] t = new KEY_TYPE[layers];

  while(stackPos > 0) {
   final int first = offsetStack[--stackPos];
   final int length = lengthStack[stackPos];
   final int level = levelStack[stackPos];
#if KEY_CLASS_Character
   final int signMask = 0;
#else
   final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
#endif

   final KEY_TYPE[] k = a[level / DIGITS_PER_ELEMENT]; // This is the key array
   final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
shift that extract the right byte from a key

   // Count keys.
   for(int i = first + length; i-- != first;) count[INT(KEY2LEXINT(k[i]) >>> shift & DIGIT_MASK ^ signMask)]++;

   // Compute cumulative distribution
   int lastUsed = -1;
   for
 (int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
    if (count[i] != 0) lastUsed = i;
```

```
   pos[i] = (p += count[i]);
  }

  final int end = first + length - count[lastUsed];
  // i moves through the start of each block
  for(int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
   for(int p = layers; p-- != 0;) t[p] = a[p][i];
   c = INT(KEY2LEXINT(k[i]) >>> shift & DIGIT_MASK ^ signMask);

   if (i < end) { // When all slots are OK, the last slot is necessarily OK.
    while((d = --pos[c]) > i) {
     c = INT(KEY2LEXINT(k[d]) >>> shift & DIGIT_MASK ^ signMask);
     for(int p = layers; p-- != 0;) {
      final KEY_TYPE u = t[p];
      t[p] = a[p][d];
      a[p][d] = u;
     }
    }
    for(int p = layers; p-- != 0;) a[p][i] = t[p];
   }

   if (level < maxLevel && count[c] > 1) {
    if (count[c] < RADIXSORT_NO_REC) selectionSort(a, i, i + count[c], level + 1);
    else {
     offsetStack[stackPos] = i;
     lengthStack[stackPos] = count[c];
     levelStack[stackPos++]
 = level + 1;
    }
   }
  }
 }
}


#endif

#endif

 /** Shuffles the specified array fragment using the specified pseudorandom number generator.
  *
  * @param a the array to be shuffled.
  * @param from the index of the first element (inclusive) to be shuffled.
  * @param to the index of the last element (exclusive) to be shuffled.
  * @param random a pseudorandom number generator.
  * @return {@code a}.
  */
 public static KEY_GENERIC KEY_GENERIC_TYPE[] shuffle(final KEY_GENERIC_TYPE[] a, final int from,
```

```
final int to, final Random random) {
 for(int i = to - from; i-- != 0;) {
  final int p = random.nextInt(i + 1);
  final KEY_GENERIC_TYPE t = a[from + i];
  a[from + i] = a[from + p];
  a[from + p] = t;
 }
 return a;
}

/** Shuffles the specified array using the specified pseudorandom number generator.
 *
 * @param a the array to be shuffled.
 * @param random a pseudorandom number generator.
 * @return {@code a}.
 */
public static KEY_GENERIC
KEY_GENERIC_TYPE[] shuffle(final KEY_GENERIC_TYPE[] a, final Random random) {
 for(int i = a.length; i-- != 0;) {
  final int p = random.nextInt(i + 1);
  final KEY_GENERIC_TYPE t = a[i];
  a[i] = a[p];
  a[p] = t;
 }
 return a;
}

/** Reverses the order of the elements in the specified array.
 *
 * @param a the array to be reversed.
 * @return {@code a}.
 */
public static KEY_GENERIC KEY_GENERIC_TYPE[] reverse(final KEY_GENERIC_TYPE[] a) {
 final int length = a.length;
 for(int i = length / 2; i-- != 0;) {
  final KEY_GENERIC_TYPE t = a[length - i - 1];
  a[length - i - 1] = a[i];
  a[i] = t;
 }
 return a;
}

/** Reverses the order of the elements in the specified array fragment.
 *
 * @param a the array to be reversed.
 * @param from the index of the first element (inclusive) to be reversed.
 * @param to the index of the last element (exclusive) to be reversed.
 * @return {@code a}.
```

```
 */
 public static KEY_GENERIC KEY_GENERIC_TYPE[] reverse(final
 KEY_GENERIC_TYPE[] a, final int from, final int to) {
  final int length = to - from;
  for(int i = length / 2; i-- != 0;) {
   final KEY_GENERIC_TYPE t = a[from + length - i - 1];
   a[from + length - i - 1] = a[from + i];
   a[from + i] = t;
  }
  return a;
 }

 /** A type-specific content-based hash strategy for arrays. */

 private static final class ArrayHashStrategy KEY_GENERIC implements
Hash.Strategy<KEY_GENERIC_TYPE[]>, java.io.Serializable {
  private static final long serialVersionUID = -7046029254386353129L;

  @Override
  public int hashCode(final KEY_GENERIC_TYPE[] o) { return java.util.Arrays.hashCode(o); }

  @Override
  public boolean equals(final KEY_GENERIC_TYPE[] a, final KEY_GENERIC_TYPE[] b) { return
java.util.Arrays.equals(a, b); }
 }

 /** A type-specific content-based hash strategy for arrays.
  *
  * <p>This hash strategy may be used in custom hash collections whenever keys are
  * arrays, and they must be considered equal by content. This strategy
  * will handle
 {@code null} correctly, and it is serializable.
  */

#if KEYS_PRIMITIVE
 public static final Hash.Strategy<KEY_TYPE[]> HASH_STRATEGY = new ArrayHashStrategy();
#else
 @SuppressWarnings({"rawtypes"})
 public static final Hash.Strategy HASH_STRATEGY = new ArrayHashStrategy();
#endif

#ifdef TEST
#if KEYS_PRIMITIVE && ! KEY_CLASS_Boolean
 private static int currentSeed = 6;

 private static final KEY_TYPE[] fillArrayForBenchmark(KEY_TYPE[] a, int uniqueValues) {
  for(int i = a.length; i-- != 0;) a[i] = (KEY_TYPE)(uniqueValues - (i % uniqueValues));
  a = shuffle(a, new Random(currentSeed++));
```

```
 return a;
}

private static final int WARMUP_RUNS = 100;
private static final int GC_EVERY = 100;
private static final int N = 1000;
private static final int UNIQUE_VALUES = 2 * N / 3;
private static final int RUNS = 1000;
public static final void main(String... args) {
 KEY_TYPE[] a = new KEY_TYPE[N];
 KEY_TYPE[] b = new KEY_TYPE[N];
 KEY_TYPE[] c = new KEY_TYPE[N];
 for (int n
= 0; n < WARMUP_RUNS; ++n) {
  a = fillArrayForBenchmark(a, UNIQUE_VALUES);
  System.arraycopy(a, 0, b, 0, N);
  System.arraycopy(a, 0, c, 0, N);
  java.util.Arrays.sort(a);
  quickSort(b);
  radixSort(c);
 }
 long arraysSortSum = 0;
 long quickSortSum = 0;
 long radixSortSum = 0;
 System.gc();
 for (int n = 0; n < RUNS; ++n) {
  a = fillArrayForBenchmark(a, UNIQUE_VALUES);
  System.arraycopy(a, 0, b, 0, N);
  System.arraycopy(a, 0, c, 0, N);

  if (n > 0 && n % GC_EVERY == 0) System.gc();
  long start = System.nanoTime();
  java.util.Arrays.sort(a);
  long end = System.nanoTime();
  arraysSortSum += end - start;

  if (n > 0 && n % GC_EVERY == 0) System.gc();
  start = System.nanoTime();
  quickSort(b);
  end = System.nanoTime();
  quickSortSum += end - start;

  if (n > 0 && n % GC_EVERY == 0) System.gc();
  start = System.nanoTime();
  radixSort(c);
  end = System.nanoTime();
  radixSortSum += end - start;
 }
```

```
  System.out.println(String.format("Quicksort
 vs radix sort for arrays of length %d with %d unique values", N, UNIQUE_VALUES));
  double averageArraysSort = (double)arraysSortSum / RUNS;
  System.out.println("java.util.Arrays.sort() average time " + averageArraysSort);
  double averageQuickSort = (double)quickSortSum / RUNS;
  System.out.println("Average quicksort average time " + averageQuickSort);
  double averageRadixSort = (double)radixSortSum / RUNS;
  System.out.println("Average radixsort average time " + averageRadixSort);
 }
#endif
#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Arrays.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2010-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

#if KEY_CLASS_Object
import java.util.Arrays;
import java.util.Comparator;
#endif

import java.io.Serializable;
import it.unimi.dsi.fastutil.HashCommon;
import it.unimi.dsi.fastutil.PriorityQueue;

import java.util.NoSuchElementException;
```

```
/** A type-specific array-based FIFO queue, supporting also deque
 operations.
*
* <p>Instances of this class represent a FIFO queue using a backing
* array in a circular way. The array is enlarged and shrunk as needed. You can use the {@link #trim()} method
* to reduce its memory usage, if necessary.
*
* <p>This class provides additional methods that implement a <em>deque</em> (double-ended queue).
*/

public class ARRAY_FIFO_QUEUE KEY_GENERIC implements PRIORITY_QUEUE KEY_GENERIC,
Serializable {
 private static final long serialVersionUID = 0L;

 /** The standard initial capacity of a queue. */
 public static final int INITIAL_CAPACITY = 4;

 /** The backing array. */
 protected transient KEY_GENERIC_TYPE array[];

 /** The current (cached) length of {@link #array}. */
 protected transient int length;

 /** The start position in {@link #array}. It is always strictly smaller than {@link #length}.*/
 protected transient int start;

 /** The end position in {@link #array}. It is always strictly smaller than {@link #length}.
  *  Might be actually
 smaller than {@link #start} because {@link #array} is used cyclically. */
 protected transient int end;

 /** Creates a new empty queue with given capacity.
  *
  * @param capacity the initial capacity of this queue.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public ARRAY_FIFO_QUEUE(final int capacity) {
  if (capacity < 0) throw new IllegalArgumentException("Initial capacity (" + capacity + ") is negative");
  array = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[Math.max(1, capacity)]; // Never build a queue with
zero-sized backing array.
  length = array.length;
 }

 /** Creates a new empty queue with standard {@linkplain #INITIAL_CAPACITY initial capacity}. */
 public ARRAY_FIFO_QUEUE() {
  this(INITIAL_CAPACITY);
 }
```

```
/** {@inheritDoc}
 * <p>This implementation returns {@code null} (FIFO queues have no comparator). */
@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() {
 return null;
}

@Override
public KEY_GENERIC_TYPE DEQUEUE() {
 if (start == end) throw new NoSuchElementException();
 final
KEY_GENERIC_TYPE t = array[start];
#if KEYS_REFERENCE
 array[start] = null; // Clean-up for the garbage collector.
#endif
 if (++start == length) start = 0;
 reduce();
 return t;
}

/** Dequeues the {@linkplain PriorityQueue#last() last} element from the queue.
 *
 * @return the dequeued element.
 * @throws NoSuchElementException if the queue is empty.
 */
public KEY_GENERIC_TYPE DEQUEUE_LAST() {
 if (start == end) throw new NoSuchElementException();
 if (end == 0) end = length;
 final KEY_GENERIC_TYPE t = array[--end];
#if KEYS_REFERENCE
 array[end] = null; // Clean-up for the garbage collector.
#endif
 reduce();
 return t;
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private final void resize(final int size, final int newLength) {
 final KEY_GENERIC_TYPE[] newArray = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[newLength];
 if (start >= end) {
  if (size != 0) {
   System.arraycopy(array, start, newArray, 0, length - start);
   System.arraycopy(array,
0, newArray, length - start, end);
  }
 }
 else System.arraycopy(array, start, newArray, 0, end - start);
```

```
 start = 0;
 end = size;
 array = newArray;
 length = newLength;
}

private final void expand() {
 resize(length, (int)Math.min(it.unimi.dsi.fastutil.Arrays.MAX_ARRAY_SIZE, 2L * length));
}

private final void reduce() {
 final int size = size();
 if (length > INITIAL_CAPACITY && size <= length / 4) resize(size, length / 2);
}

@Override
public void enqueue(KEY_GENERIC_TYPE x) {
 array[end++] = x;
 if (end == length) end = 0;
 if (end == start) expand();
}

/** Enqueues a new element as the first element (in dequeuing order) of the queue.
 * @param x the element to enqueue.
 */
public void enqueueFirst(KEY_GENERIC_TYPE x) {
 if (start == 0) start = length;
 array[--start] = x;
 if (end == start) expand();
}

@Override
public KEY_GENERIC_TYPE FIRST() {
 if (start == end) throw new NoSuchElementException();
 return array[start];
}


@Override
public
KEY_GENERIC_TYPE LAST() {
 if (start == end) throw new NoSuchElementException();
 return array[(end == 0 ? length : end) - 1];
}

@Override
public void clear() {
#if KEYS_REFERENCE
```

```
 if (start <= end) Arrays.fill(array, start, end, null);
 else {
  Arrays.fill(array, start, length, null);
  Arrays.fill(array, 0, end, null);
 }
#endif
 start = end = 0;
}

/** Trims the queue to the smallest possible size. */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public void trim() {
 final int size = size();
 final KEY_GENERIC_TYPE[] newArray =
#if KEYS_PRIMITIVE
        new KEY_GENERIC_TYPE[size + 1];
#else
        (KEY_GENERIC_TYPE[])new Object[size + 1];
#endif
 if (start <= end) System.arraycopy(array, start, newArray, 0, end - start);
 else {
  System.arraycopy(array, start, newArray, 0, length - start);
  System.arraycopy(array, 0, newArray, length - start, end);
 }
 start = 0;
 length = (end = size) + 1;
 array = newArray;
}

@Override
public int size() {
 final int
apparentLength = end - start;
 return apparentLength >= 0 ? apparentLength : length + apparentLength;
}

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
 s.defaultWriteObject();
 int size = size();
 s.writeInt(size);
 for(int i = start; size-- != 0;) {
  s.WRITE_KEY(array[i++]);
  if (i == length) i = 0;
 }
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
```

```
  s.defaultReadObject();
  end = s.readInt();
  array = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[length = HashCommon.nextPowerOfTwo(end + 1)];
  for(int i = 0; i < end; i++) array[i] = KEY_GENERIC_CAST s.READ_KEY();
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/ArrayFIFOQueue.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

#if KEYS_REFERENCE
import it.unimi.dsi.fastutil.Stack;
#endif

import java.util.List;
import java.util.Iterator;
import java.util.ListIterator;
import java.util.Collection;
import java.util.NoSuchElementException;

/**  An abstract class providing basic methods for lists implementing a type-specific
 list interface.
*
* <p>As an additional bonus, this class implements on top of the list operations a type-specific stack.
*/

public abstract class ABSTRACT_LIST KEY_GENERIC extends ABSTRACT_COLLECTION KEY_GENERIC
```

```
implements LIST KEY_GENERIC, STACK KEY_GENERIC {

 protected ABSTRACT_LIST() {}

 /** Ensures that the given index is nonnegative and not greater than the list size.
  *
  * @param index an index.
  * @throws IndexOutOfBoundsException if the given index is negative or greater than the list size.
  */
 protected void ensureIndex(final int index) {
  if (index < 0)  throw new IndexOutOfBoundsException("Index (" + index + ") is negative");
  if (index > size()) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than list size (" +
(size()) + ")");
 }

 /** Ensures that the given index is nonnegative and smaller than the list size.
  *
  * @param index an index.
  * @throws IndexOutOfBoundsException if the given index is negative or not smaller than the list
 size.
  */
 protected void ensureRestrictedIndex(final int index) {
  if (index < 0)  throw new IndexOutOfBoundsException("Index (" + index + ") is negative");
  if (index >= size()) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list
size (" + (size()) + ")");
 }

 /** {@inheritDoc}
  *
  * <p>This implementation always throws an {@link UnsupportedOperationException}.
  */
 @Override
 public void add(final int index, final KEY_GENERIC_TYPE k) {
  throw new UnsupportedOperationException();
 }

 /** {@inheritDoc}
  *
  * <p>This implementation delegates to the type-specific version of {@link List#add(int, Object)}.
  */
 @Override
 public boolean add(final KEY_GENERIC_TYPE k) {
  add(size(), k);
  return true;
 }

 /** {@inheritDoc}
  *
```

```
 * <p>This implementation always throws an {@link UnsupportedOperationException}.
 */
@Override
public KEY_GENERIC_TYPE REMOVE_KEY(final int i) {
 throw new UnsupportedOperationException();
}

/** {@inheritDoc}

*
 * <p>This implementation always throws an {@link UnsupportedOperationException}.
 */
@Override
public KEY_GENERIC_TYPE set(final int index, final KEY_GENERIC_TYPE k) {
 throw new UnsupportedOperationException();
}

/** Adds all of the elements in the specified collection to this list (optional operation). */
@Override
public boolean addAll(int index, final Collection<? extends KEY_GENERIC_CLASS> c) {
 ensureIndex(index);
 final Iterator<? extends KEY_GENERIC_CLASS> i = c.iterator();
 final boolean retVal = i.hasNext();
 while(i.hasNext()) add(index++, KEY_CLASS2TYPE(i.next()));
 return retVal;
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link List#addAll(int, Collection)}.
 */
@Override
public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) {
 return addAll(size(), c);
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to {@link #listIterator()}.
 */
@Override
public KEY_LIST_ITERATOR
KEY_GENERIC iterator() {
 return listIterator();
}

/** {@inheritDoc}
 *
```

```
 * <p>This implementation delegates to {@link #listIterator(int) listIterator(0)}.
 */
@Override
public KEY_LIST_ITERATOR KEY_GENERIC listIterator() {
 return listIterator(0);
}

/** {@inheritDoc}
 * <p>This implementation is based on the random-access methods. */
@Override
public KEY_LIST_ITERATOR KEY_GENERIC listIterator(final int index) {
 ensureIndex(index);

 return new KEY_LIST_ITERATOR KEY_GENERIC() {
  int pos = index, last = -1;

  @Override
  public boolean hasNext() { return pos < ABSTRACT_LIST.this.size(); }
  @Override
  public boolean hasPrevious() { return pos > 0; }
  @Override
  public KEY_GENERIC_TYPE NEXT_KEY() { if (! hasNext()) throw new NoSuchElementException(); return
ABSTRACT_LIST.this.GET_KEY(last = pos++); }
  @Override
  public KEY_GENERIC_TYPE PREV_KEY() { if (! hasPrevious()) throw new NoSuchElementException(); return
ABSTRACT_LIST.this.GET_KEY(last
 = --pos); }
  @Override
  public int nextIndex() { return pos; }
  @Override
  public int previousIndex() { return pos - 1; }
  @Override
  public void add(final KEY_GENERIC_TYPE k) {
   ABSTRACT_LIST.this.add(pos++, k);
   last = -1;
  }
  @Override
  public void set(final KEY_GENERIC_TYPE k) {
   if (last == -1) throw new IllegalStateException();
   ABSTRACT_LIST.this.set(last, k);
  }
  @Override
  public void remove() {
   if (last == -1) throw new IllegalStateException();
   ABSTRACT_LIST.this.REMOVE_KEY(last);
   /* If the last operation was a next(), we are removing an element *before* us, and we must decrease pos
correspondingly. */
   if (last < pos) pos--;
```

```
    last = -1;
   }
  };
}

/** Returns true if this list contains the specified element.
 * <p>This implementation delegates to {@code indexOf()}.
 * @see List#contains(Object)
 */

@Override
public boolean contains(final KEY_TYPE k) {
 return indexOf(k) >= 0;
}

@Override
public
int indexOf(final KEY_TYPE k) {
 final KEY_LIST_ITERATOR KEY_GENERIC i = listIterator();
 KEY_GENERIC_TYPE e;
 while(i.hasNext()) {
  e = i.NEXT_KEY();
  if (KEY_EQUALS(k, e)) return i.previousIndex();
 }
 return -1;
}

@Override
public int lastIndexOf(final KEY_TYPE k) {
 KEY_LIST_ITERATOR KEY_GENERIC i = listIterator(size());
 KEY_GENERIC_TYPE e;
 while(i.hasPrevious()) {
  e = i.PREV_KEY();
  if (KEY_EQUALS(k, e)) return i.nextIndex();
 }
 return -1;
}

@Override
public void size(final int size) {
 int i = size();
 if (size > i) while(i++ < size) add(KEY_NULL);
 else while(i-- != size) REMOVE_KEY(i);
}


@Override
public LIST KEY_GENERIC subList(final int from, final int to) {
```

```
 ensureIndex(from);
 ensureIndex(to);
 if (from > to) throw new IndexOutOfBoundsException("Start index (" + from + ") is greater than end index (" + to +
")");

 return new SUBLIST KEY_GENERIC_DIAMOND(this, from, to);
}

/** {@inheritDoc}
 *
 * <p>This is a
trivial iterator-based implementation. It is expected that
 * implementations will override this method with a more optimized version.
 */
@Override
public void removeElements(final int from, final int to) {
 ensureIndex(to);
 KEY_LIST_ITERATOR KEY_GENERIC i = listIterator(from);
 int n = to - from;
 if (n < 0) throw new IllegalArgumentException("Start index (" + from + ") is greater than end index (" + to + ")");
 while(n-- != 0) {
  i.NEXT_KEY();
  i.remove();
 }
}

/** {@inheritDoc}
 *
 * <p>This is a trivial iterator-based implementation. It is expected that
 * implementations will override this method with a more optimized version.
 */
@Override
public void addElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) {
 ensureIndex(index);
 if (offset < 0) throw new ArrayIndexOutOfBoundsException("Offset (" + offset + ") is negative");
 if (offset + length > a.length) throw new ArrayIndexOutOfBoundsException("End index (" + (offset + length) +
") is greater than array length (" + a.length + ")");
 while(length-- != 0) add(index++, a[offset++]);
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the analogous method for array fragments.
 */
@Override
public void addElements(final int index, final KEY_GENERIC_TYPE a[]) {
 addElements(index, a, 0, a.length);
}
```

```
 /** {@inheritDoc}
  *
  * <p>This is a trivial iterator-based implementation. It is expected that
  * implementations will override this method with a more optimized version.
  */
 @Override
 public void getElements(final int from, final KEY_TYPE a[], int offset, int length) {
  KEY_LIST_ITERATOR KEY_GENERIC i = listIterator(from);
  if (offset < 0) throw new ArrayIndexOutOfBoundsException("Offset (" + offset + ") is negative");
  if (offset + length > a.length) throw new ArrayIndexOutOfBoundsException("End index (" + (offset + length) + ") is
greater than array length (" + a.length + ")");
  if (from + length > size()) throw new IndexOutOfBoundsException("End
 index (" + (from + length) + ") is greater than list size (" + size() + ")");
  while(length-- != 0) a[offset++] = i.NEXT_KEY();
 }

 /** {@inheritDoc}
  * <p>This implementation delegates to {@link #removeElements(int, int)}.*/
 @Override
 public void clear() {
  removeElements(0, size());
 }


#if ! KEY_CLASS_Reference
 private boolean valEquals(final Object a, final Object b) {
  return a == null ? b == null : a.equals(b);
 }
#endif

 /** Returns the hash code for this list, which is identical to {@link java.util.List#hashCode()}.
  *
  * @return the hash code for this list.
  */
 @Override
 public int hashCode() {
  KEY_ITERATOR KEY_GENERIC i = iterator();
  int h = 1, s = size();
  while (s-- != 0) {
   KEY_GENERIC_TYPE k = i.NEXT_KEY();
   h = 31 * h + KEY2JAVAHASH(k);
  }
  return h;
 }


 @Override
```

```
 public boolean equals(final Object o) {
  if (o == this) return true;
  if (! (o instanceof List)) return false;

  final List<?> l = (List<?>)o;
  int s = size();
  if
 (s != l.size()) return false;

#if KEYS_PRIMITIVE
  if (l instanceof LIST) {
   final KEY_LIST_ITERATOR KEY_GENERIC i1 = listIterator(), i2 = ((LIST KEY_GENERIC)l).listIterator();
   while(s-- != 0) if (i1.NEXT_KEY() != i2.NEXT_KEY()) return false;
   return true;
  }
#endif

  final ListIterator<?> i1 = listIterator(), i2 = l.listIterator();

#if KEY_CLASS_Reference
  while(s-- !=  0) if (i1.next() != i2.next()) return false;
#else
  while(s-- !=  0) if (! valEquals(i1.next(), i2.next())) return false;
#endif
  return true;
 }

#if ! KEY_CLASS_Reference
 /** Compares this list to another object. If the
  * argument is a {@link java.util.List}, this method performs a lexicographical comparison; otherwise,
  * it throws a {@code ClassCastException}.
  *
  * @param l a list.
  * @return if the argument is a {@link java.util.List}, a negative integer,
  * zero, or a positive integer as this list is lexicographically less than, equal
  * to, or greater than the argument.
  * @throws ClassCastException
 if the argument is not a list.
  */

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 @Override
 public int compareTo(final List<? extends KEY_GENERIC_CLASS> l) {
  if (l == this) return 0;

  if (l instanceof LIST) {

   final KEY_LIST_ITERATOR KEY_GENERIC i1 = listIterator(), i2 = ((LIST KEY_GENERIC)l).listIterator();
```

```
  int r;
  KEY_GENERIC_TYPE e1, e2;

  while(i1.hasNext() && i2.hasNext()) {
   e1 = i1.NEXT_KEY();
   e2 = i2.NEXT_KEY();
   if ((r = KEY_CMP(e1, e2)) != 0) return r;
  }
  return i2.hasNext() ? -1 : (i1.hasNext() ? 1 : 0);
 }

 ListIterator<? extends KEY_GENERIC_CLASS> i1 = listIterator(), i2 = l.listIterator();
 int r;

 while(i1.hasNext() && i2.hasNext()) {
  if ((r = ((Comparable<? super KEY_GENERIC_CLASS>)i1.next()).compareTo(i2.next())) != 0) return r;
 }
 return i2.hasNext() ? -1 : (i1.hasNext() ? 1 : 0);
}
#endif

@Override
public void push(final KEY_GENERIC_TYPE o) {
 add(o);
}

@Override
public KEY_GENERIC_TYPE POP() {
 if (isEmpty()) throw
new NoSuchElementException();
 return REMOVE_KEY(size() - 1);
}

@Override
public KEY_GENERIC_TYPE TOP() {
 if (isEmpty()) throw new NoSuchElementException();
 return GET_KEY(size() - 1);
}

@Override
public KEY_GENERIC_TYPE PEEK(final int i) {
 return GET_KEY(size() - 1 - i);
}

#if KEYS_PRIMITIVE

/** Removes a single instance of the specified element from this collection, if it is present (optional operation).
 * <p>This implementation delegates to {@code indexOf()}.
```

```
 * @see List#remove(Object)
 */
@Override
public boolean rem(final KEY_TYPE k) {
 int index = indexOf(k);
 if (index == -1) return false;
 REMOVE_KEY(index);
 return true;
}

@Override
public boolean addAll(int index, final COLLECTION c) {
 ensureIndex(index);
 final KEY_ITERATOR i = c.iterator();
 final boolean retVal = i.hasNext();
 while(i.hasNext()) add(index++, i.NEXT_KEY());
 return retVal;
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version
of {@link List#addAll(int, Collection)}.
 */
@Override
public boolean addAll(final int index, final LIST l) {
 return addAll(index, (COLLECTION)l);
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link List#addAll(int, Collection)}.
 */
@Override
public boolean addAll(final COLLECTION c) {
 return addAll(size(), c);
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific list version of {@link List#addAll(int, Collection)}.
 */
@Override
public boolean addAll(final LIST l) {
 return addAll(size(), l);
}

#endif
```

```
 @Override
 public String toString() {
  final StringBuilder s = new StringBuilder();
  final KEY_ITERATOR KEY_GENERIC i = iterator();
  int n = size();
  KEY_GENERIC_TYPE k;
  boolean first = true;

  s.append("[");

  while(n-- != 0) {
   if (first) first = false;
   else s.append(", ");
   k = i.NEXT_KEY();
#if KEYS_REFERENCE
   if (this == k) s.append("(this list)"); else
#endif
    s.append(String.valueOf(k));
  }

  s.append("]");
  return
 s.toString();
 }

 /** A class implementing a sublist view. */
 public static class SUBLIST KEY_GENERIC extends ABSTRACT_LIST KEY_GENERIC implements
java.io.Serializable {
  private static final long serialVersionUID = -7046029254386353129L;
  /** The list this sublist restricts. */
  protected final LIST KEY_GENERIC l;
  /** Initial (inclusive) index of this sublist. */
  protected final int from;
  /** Final (exclusive) index of this sublist. */
  protected int to;

  public SUBLIST(final LIST KEY_GENERIC l, final int from, final int to) {
   this.l = l;
   this.from = from;
   this.to = to;
  }

  private boolean assertRange() {
   assert from <= l.size();
   assert to <= l.size();
   assert to >= from;
   return true;
```

```
 }

 @Override
 public boolean add(final KEY_GENERIC_TYPE k) {
  l.add(to, k);
  to++;
  assert assertRange();
  return true;
 }

 @Override
 public void add(final int index, final KEY_GENERIC_TYPE
k) {
  ensureIndex(index);
  l.add(from + index, k);
  to++;
  assert assertRange();
 }

 @Override
 public boolean addAll(final int index, final Collection<? extends KEY_GENERIC_CLASS> c) {
  ensureIndex(index);
  to += c.size();
  return l.addAll(from + index, c);
 }

 @Override
 public KEY_GENERIC_TYPE GET_KEY(final int index) {
  ensureRestrictedIndex(index);
  return l.GET_KEY(from + index);
 }

 @Override
 public KEY_GENERIC_TYPE REMOVE_KEY(final int index) {
  ensureRestrictedIndex(index);
  to--;
  return l.REMOVE_KEY(from + index);
 }

 @Override
 public KEY_GENERIC_TYPE set(final int index, final KEY_GENERIC_TYPE k) {
  ensureRestrictedIndex(index);
  return l.set(from + index, k);
 }

 @Override
 public int size() {
  return to - from;
```

```
 }

 @Override
 public void getElements(final int from, final KEY_TYPE[] a, final int offset, final int length) {
  ensureIndex(from);
  if (from + length > size())  throw new IndexOutOfBoundsException("End
index (" + from + length + ") is greater than list size (" + size() + ")");
  l.getElements(this.from + from, a, offset, length);
 }

 @Override
 public void removeElements(final int from, final int to) {
  ensureIndex(from);
  ensureIndex(to);
  l.removeElements(this.from + from, this.from + to);
  this.to -= (to - from);
  assert assertRange();
 }

 @Override
 public void addElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) {
  ensureIndex(index);
  l.addElements(this.from + index, a, offset, length);
  this.to += length;
  assert assertRange();
 }

 @Override
 public KEY_LIST_ITERATOR KEY_GENERIC listIterator(final int index) {
  ensureIndex(index);

  return new KEY_LIST_ITERATOR KEY_GENERIC() {
    int pos = index, last = -1;

    @Override
    public boolean hasNext() { return pos < size(); }
    @Override
    public boolean hasPrevious() { return pos > 0; }
    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() { if (! hasNext()) throw
new NoSuchElementException(); return l.GET_KEY(from + (last = pos++)); }
    @Override
    public KEY_GENERIC_TYPE PREV_KEY() { if (! hasPrevious()) throw new NoSuchElementException(); return
l.GET_KEY(from + (last = --pos)); }
    @Override
    public int nextIndex() { return pos; }
    @Override
    public int previousIndex() { return pos - 1; }
```

```
    @Override
    public void add(KEY_GENERIC_TYPE k) {
     if (last == -1) throw new IllegalStateException();
     SUBLIST.this.add(pos++, k);
     last = -1;
     assert assertRange();
    }
    @Override
    public void set(KEY_GENERIC_TYPE k) {
     if (last == -1) throw new IllegalStateException();
     SUBLIST.this.set(last, k);
    }
    @Override
    public void remove() {
     if (last == -1) throw new IllegalStateException();
     SUBLIST.this.REMOVE_KEY(last);
     /* If the last operation was a next(), we are removing an element *before* us, and we must decrease pos
correspondingly. */
     if (last <
 pos) pos--;
     last = -1;
     assert assertRange();
    }
   };
  }

  @Override
  public LIST KEY_GENERIC subList(final int from, final int to) {
   ensureIndex(from);
   ensureIndex(to);
   if (from > to) throw new IllegalArgumentException("Start index (" + from + ") is greater than end index (" + to +
")");

   return new SUBLIST KEY_GENERIC_DIAMOND(this, from, to);
  }

#if KEYS_PRIMITIVE

  @Override
  public boolean rem(final KEY_TYPE k) {
   int index = indexOf(k);
   if (index == -1) return false;
   to--;
   l.REMOVE_KEY(from + index);
   assert assertRange();
   return true;
  }
```

```
  @Override
  public boolean addAll(final int index, final COLLECTION c) {
   ensureIndex(index);
   return super.addAll(index, c);
  }

  @Override
  public boolean addAll(final int index, final LIST l) {
   ensureIndex(index);
   return super.addAll(index, l);
  }
#endif

 }

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractList.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

/**  An abstract class facilitating the creation of type-specific {@linkplain it.unimi.dsi.fastutil.BigListIterator big-list iterators}.
*
* @deprecated As of fastutil 8 this class is no longer necessary, as its previous abstract
* methods are now default methods of the type-specific interface.
*/
```

```
@Deprecated
public
 abstract class KEY_ABSTRACT_BIG_LIST_ITERATOR KEY_GENERIC extends
KEY_ABSTRACT_BIDI_ITERATOR KEY_GENERIC implements KEY_BIG_LIST_ITERATOR
KEY_GENERIC {
 protected KEY_ABSTRACT_BIG_LIST_ITERATOR() {}
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractBigListIterator.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.SortedSet;
import java.util.Collection;

/** A type-specific {@link SortedSet}; provides some additional methods that use polymorphism to avoid
(un)boxing.
*
* <p>Additionally, this interface strengthens {@link #iterator()},
* {@link #comparator()} (for primitive types), {@link SortedSet#subSet(Object,Object)},
 * {@link SortedSet#headSet(Object)} and {@link SortedSet#tailSet(Object)}.
*
* @see SortedSet
*/


public interface SORTED_SET KEY_GENERIC extends SET KEY_GENERIC,
SortedSet<KEY_GENERIC_CLASS>, KEY_BIDI_ITERABLE KEY_GENERIC {
```

```
/** Returns a type-specific {@link it.unimi.dsi.fastutil.BidirectionalIterator} on the elements in
 * this set, starting from a given element of the domain (optional operation).
 *
 * <p>This method returns a type-specific bidirectional iterator with given
 * starting point. The starting point is any element comparable to the
 * elements of this set (even if it does not actually belong to the
 * set). The next element of the returned iterator is the least element of
 * the set that is greater than the starting point (if there are no
 * elements greater than the starting point, {@link
 * it.unimi.dsi.fastutil.BidirectionalIterator#hasNext() hasNext()} will return
 * {@code false}). The previous element of the returned
iterator is
 * the greatest element of the set that is smaller than or equal to the
 * starting point (if there are no elements smaller than or equal to the
 * starting point, {@link it.unimi.dsi.fastutil.BidirectionalIterator#hasPrevious()
 * hasPrevious()} will return {@code false}).
 *
 * <p>Note that passing the last element of the set as starting point and
 * calling {@link it.unimi.dsi.fastutil.BidirectionalIterator#previous() previous()} you can traverse the
 * entire set in reverse order.
 *
 * @param fromElement an element to start from.
 * @return a bidirectional iterator on the element in this set, starting at the given element.
 * @throws UnsupportedOperationException if this set does not support iterators with a starting point.
 */

KEY_BIDI_ITERATOR KEY_GENERIC iterator(KEY_GENERIC_TYPE fromElement);

/** Returns a type-specific {@link it.unimi.dsi.fastutil.BidirectionalIterator} on the elements in
 * this set.
 *
 * <p>Note that this specification strengthens
the one given in the corresponding type-specific
 * {@link Collection}.
 *
 * <p>This method returns a parameterised bidirectional iterator. The iterator
 * can be moreover safely cast to a type-specific iterator.
 *
 * @return a bidirectional iterator on the element in this set.
 */

@Override
KEY_BIDI_ITERATOR KEY_GENERIC iterator();

/** Returns a view of the portion of this sorted set whose elements range from {@code fromElement}, inclusive, to
{@code toElement}, exclusive.
 * <p>Note that this specification strengthens the one given in {@link SortedSet#subSet(Object,Object)}.
```

```
  * @see SortedSet#subSet(Object,Object)
  */

#if KEYS_REFERENCE
 @Override
#endif
 SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_TYPE fromElement, KEY_GENERIC_TYPE
toElement) ;

 /** Returns a view of the portion of this sorted set whose elements are strictly less than {@code toElement}.
  * <p>Note that this specification strengthens the one given in {@link SortedSet#headSet(Object)}.
  * @see
 SortedSet#headSet(Object)
  */

#if KEYS_REFERENCE
 @Override
#endif
 SORTED_SET KEY_GENERIC headSet(KEY_GENERIC_TYPE toElement);

 /** Returns a view of the portion of this sorted set whose elements are greater than or equal to {@code
fromElement}.
  * <p>Note that this specification strengthens the one given in {@link SortedSet#headSet(Object)}.
  * @see SortedSet#tailSet(Object)
  */

#if KEYS_REFERENCE
 @Override
#endif
 SORTED_SET KEY_GENERIC tailSet(KEY_GENERIC_TYPE fromElement);

#if KEYS_PRIMITIVE
 /** {@inheritDoc}
  * <p>Note that this specification strengthens the one given in {@link SortedSet#comparator()}.
  */
 @Override
 KEY_COMPARATOR comparator();

 /** Returns the first (lowest) element currently in this set.
  * @see SortedSet#first()
  */
 KEY_TYPE FIRST();

 /** Returns the last (highest) element currently in this set.
  * @see SortedSet#last()
  */
 KEY_TYPE LAST();
```

```
 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead.

 */
 @Deprecated
 @Override
 default SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_CLASS from, final
KEY_GENERIC_CLASS to) {
  return subSet(from.KEY_VALUE(), to.KEY_VALUE());
 }

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead.
  */
 @Deprecated
 @Override
 default SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_CLASS to) {
  return headSet(to.KEY_VALUE());
 }

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead.
  */
 @Deprecated
 @Override
 default SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_CLASS from) {
  return tailSet(from.KEY_VALUE());
 }

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead.
  */
 @Deprecated
 @Override
 default KEY_GENERIC_CLASS first() {
  return KEY2OBJ(FIRST());
 }

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead.
  */
 @Deprecated
 @Override
 default KEY_CLASS last() {
  return KEY2OBJ(LAST());
 }

#endif
```

}

Found
 in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/SortedSet.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.Collection;

/** A type-specific {@link Collection}; provides some additional methods
* that use polymorphism to avoid (un)boxing.
*
* <p>Additionally, this class defines strengthens (again) {@link #iterator()}.
*
* @see Collection
*/

public interface COLLECTION KEY_GENERIC extends
 Collection<KEY_GENERIC_CLASS>, KEY_ITERABLE KEY_GENERIC {

 /** Returns a type-specific iterator on the elements of this collection.
  *
  * <p>Note that this specification strengthens the one given in
  * {@link java.lang.Iterable#iterator()}, which was already
  * strengthened in the corresponding type-specific class,
  * but was weakened by the fact that this interface extends {@link Collection}.
  *
```

```
 * @return a type-specific iterator on the elements of this collection.
 */
@Override
KEY_ITERATOR KEY_GENERIC iterator();

#if KEYS_PRIMITIVE

/** Ensures that this collection contains the specified element (optional operation).
 * @see Collection#add(Object)
 */
boolean add(KEY_TYPE key);

/** Returns {@code true} if this collection contains the specified element.
 * @see Collection#contains(Object)
 */
boolean contains(KEY_TYPE key);

/** Removes a single instance of the specified element from this
 * collection, if it is present (optional operation).
 *
 * <p>Note
that this method should be called {@link java.util.Collection#remove(Object) remove()}, but the clash
 * with the similarly named index-based method in the {@link java.util.List} interface
 * forces us to use a distinguished name. For simplicity, the set interfaces reinstates
 * {@code remove()}.
 *
 * @see Collection#remove(Object)
 */
boolean rem(KEY_TYPE key);

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default boolean add(final KEY_GENERIC_CLASS key) {
 return add(KEY_CLASS2TYPE(key));
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default boolean contains(final Object key) {
 if (key == null) return false;
 return contains(KEY_OBJ2TYPE(key));
}
```

```
/** {@inheritDoc}
 * @deprecated Please use (and implement) the {@code rem()} method instead.
 */
@Deprecated
@Override
default boolean remove(final Object key)
{
 if (key == null) return false;
 return rem(KEY_OBJ2TYPE(key));
}

/** Returns a primitive type array containing the items of this collection.
 * @return a primitive type array containing the items of this collection.
 * @see Collection#toArray()
 */
KEY_TYPE[] TO_KEY_ARRAY();

/** Returns a primitive type array containing the items of this collection.
 *
 * <p>Note that, contrarily to {@link Collection#toArray(Object[])}, this
 * methods just writes all elements of this collection: no special
 * value will be added after the last one.
 *
 * @param a if this array is big enough, it will be used to store this collection.
 * @return a primitive type array containing the items of this collection.
 * @see Collection#toArray(Object[])
 * @deprecated Please use {@code toArray()} instead&mdash;this method is redundant and will be removed in the
future.
 */
@Deprecated
KEY_TYPE[] TO_KEY_ARRAY(KEY_TYPE a[]);

/** Returns an array containing all of the elements in this
collection; the runtime type of the returned array is that of the specified array.
 *
 * <p>Note that, contrarily to {@link Collection#toArray(Object[])}, this
 * methods just writes all elements of this collection: no special
 * value will be added after the last one.
 *
 * @param a if this array is big enough, it will be used to store this collection.
 * @return a primitive type array containing the items of this collection.
 * @see Collection#toArray(Object[])
 */
KEY_TYPE[] toArray(KEY_TYPE a[]);

/** Adds all elements of the given type-specific collection to this collection.
 *
```

```
 * @param c a type-specific collection.
 * @see Collection#addAll(Collection)
 * @return {@code true} if this collection changed as a result of the call.
 */
boolean addAll(COLLECTION c);

/** Checks whether this collection contains all elements from the given type-specific collection.
 *
 * @param c a type-specific collection.
 * @see Collection#containsAll(Collection)
 * @return {@code
true} if this collection contains all elements of the argument.
 */
boolean containsAll(COLLECTION c);

/** Remove from this collection all elements in the given type-specific collection.
 *
 * @param c a type-specific collection.
 * @see Collection#removeAll(Collection)
 * @return {@code true} if this collection changed as a result of the call.
 */
boolean removeAll(COLLECTION c);

#ifdef JDK_PRIMITIVE_PREDICATE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default boolean removeIf(final java.util.function.Predicate<? super KEY_GENERIC_CLASS> filter) {
 return removeIf((JDK_PRIMITIVE_PREDICATE) key -> filter.test(KEY2OBJ(KEY_NARROWING(key))));
}

/** Remove from this collection all elements which satisfy the given predicate.
 *
 * @param filter a predicate which returns {@code true} for elements to be
 *       removed.
 * @see Collection#removeIf(java.util.function.Predicate)
 * @return
{@code true} if any elements were removed.
 */
@SuppressWarnings("overloads")
default boolean removeIf(final JDK_PRIMITIVE_PREDICATE filter) {
 java.util.Objects.requireNonNull(filter);
 boolean removed = false;
 final KEY_ITERATOR each = iterator();
 while (each.hasNext()) {
  if (filter.test(each.NEXT_KEY())) {
```

```
    each.remove();
    removed = true;
   }
  }
  return removed;
 }
#endif

 /** Retains in this collection only elements from the given type-specific collection.
  *
  * @param c a type-specific collection.
  * @see Collection#retainAll(Collection)
  * @return {@code true} if this collection changed as a result of the call.
  */
 boolean retainAll(COLLECTION c);

#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Collection.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

/** A class providing static methods and objects that do useful things with type-specific priority queues.
*
* @see it.unimi.dsi.fastutil.PriorityQueue
*/
```

```
public final class PRIORITY_QUEUES {

 private PRIORITY_QUEUES() {}

 /** A synchronized wrapper class for priority queues. */

 public static
 class SynchronizedPriorityQueue KEY_GENERIC implements PRIORITY_QUEUE KEY_GENERIC {
  protected final PRIORITY_QUEUE KEY_GENERIC q;
  protected final Object sync;

  protected SynchronizedPriorityQueue(final PRIORITY_QUEUE KEY_GENERIC q, final Object sync) {
   this.q = q;
   this.sync = sync;
  }

  protected SynchronizedPriorityQueue(final PRIORITY_QUEUE KEY_GENERIC q) {
   this.q = q;
   this.sync = this;
  }

  @Override
  public void enqueue(KEY_GENERIC_TYPE x) { synchronized(sync) { q.enqueue(x); } }

  @Override
  public KEY_GENERIC_TYPE DEQUEUE() { synchronized(sync) { return q.DEQUEUE(); } }

  @Override
  public KEY_GENERIC_TYPE FIRST() { synchronized(sync) { return q.FIRST(); } }

  @Override
  public KEY_GENERIC_TYPE LAST() { synchronized(sync) { return q.LAST(); } }

  @Override
  public boolean isEmpty() { synchronized(sync) { return q.isEmpty(); } }

  @Override
  public int size() { synchronized(sync) { return q.size(); } }

  @Override
  public void clear() { synchronized(sync)
 { q.clear(); } }

  @Override
  public void changed() { synchronized(sync) { q.changed(); } }

  @Override
```

```
  public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { synchronized(sync) { return
q.comparator(); } }

#if KEYS_PRIMITIVE

  @Deprecated
  @Override
  public void enqueue(KEY_CLASS x) { synchronized(sync) { q.enqueue(x); } }

  @Deprecated
  @Override
  public KEY_CLASS dequeue() { synchronized(sync) { return q.dequeue(); } }

  @Deprecated
  @Override
  public KEY_CLASS first() { synchronized(sync) { return q.first(); } }

  @Deprecated
  @Override
  public KEY_CLASS last() { synchronized(sync) { return q.last(); } }

#endif
  @Override
  public int hashCode() { synchronized(sync) { return q.hashCode(); } }

  @Override
  public boolean equals(final Object o) { if (o == this) return true; synchronized(sync) { return q.equals(o); } }

  private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
   synchronized(sync) { s.defaultWriteObject();
 }
  }
 }


 /** Returns a synchronized type-specific priority queue backed by the specified type-specific priority queue.
  *
  * @param q the priority queue to be wrapped in a synchronized priority queue.
  * @return a synchronized view of the specified priority queue.
  */
 public static KEY_GENERIC PRIORITY_QUEUE KEY_GENERIC synchronize(final PRIORITY_QUEUE
KEY_GENERIC q) { return new SynchronizedPriorityQueue(q); }

 /** Returns a synchronized type-specific priority queue backed by the specified type-specific priority queue, using
an assigned object to synchronize.
  *
  * @param q the priority queue to be wrapped in a synchronized priority queue.
  * @param sync an object that will be used to synchronize the access to the priority queue.
```

```
  * @return a synchronized view of the specified priority queue.
  */

 public static KEY_GENERIC PRIORITY_QUEUE KEY_GENERIC synchronize(final PRIORITY_QUEUE
KEY_GENERIC q, final Object sync) { return new SynchronizedPriorityQueue(q, sync); }


}
```

Found
 in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/PriorityQueues.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

/**  An abstract class providing basic methods for priority queues implementing a type-specific interface.
* @deprecated As of fastutil 8 this class is no longer necessary, as its previous abstract
* methods are now default methods of the type-specific interface.
*/
@Deprecated
public abstract class
 ABSTRACT_PRIORITY_QUEUE KEY_GENERIC extends
it.unimi.dsi.fastutil.AbstractPriorityQueue<KEY_GENERIC_CLASS> implements java.io.Serializable,
PRIORITY_QUEUE KEY_GENERIC  {
 private static final long serialVersionUID = 1L;
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-

gz/CloudburstMC-fastutil-6e390a2/drv/AbstractPriorityQueue.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/



/** Loads elements from a given fast buffered reader, storing them in a given array fragment.
*
* @param reader a buffered reader.
* @param array an array which will be filled with data from {@code reader}.
* @param offset the index of the first element of {@code array} to be filled.
* @param length the number of
 elements of {@code array} to be filled.
* @return the number of elements actually read from {@code reader} (it might be less than {@code length} if
{@code reader} ends).
*/
public static int LOAD_KEYS(final BufferedReader reader, final KEY_TYPE[] array, final int offset, final int
length) throws IOException {
 PACKAGE.ARRAYS.ensureOffsetLength(array, offset, length);
 int i = 0;
 String s;
 try {
  for(i = 0; i < length; i++)
   if ((s = reader.readLine()) != null) array[i + offset] = KEY_CLASS.PARSE_KEY(s.trim());
   else break;
 }
 catch(EOFException itsOk) {}
 return i;
}

/** Loads elements from a given buffered reader, storing them in a given array.
*
* @param reader a buffered reader.
```

```
* @param array an array which will be filled with data from {@code reader}.
* @return the number of elements actually read from {@code reader} (it might be less than the array length if
{@code reader} ends).
*/
public static int LOAD_KEYS(final BufferedReader reader, final KEY_TYPE[] array)
 throws IOException {
 return LOAD_KEYS(reader, array, 0, array.length);
}

/** Loads elements from a file given by a {@link File} object, storing them in a given array fragment.
*
* @param file a file.
* @param array an array which will be filled with data from the specified file.
* @param offset the index of the first element of {@code array} to be filled.
* @param length the number of elements of {@code array} to be filled.
* @return the number of elements actually read from the given file (it might be less than {@code length} if the file
is too short).
*/
public static int LOAD_KEYS(final File file, final KEY_TYPE[] array, final int offset, final int length) throws
IOException {
 final BufferedReader reader = new BufferedReader(new FileReader(file));
 final int result = LOAD_KEYS(reader, array, offset, length);
 reader.close();

 return result;
}

/** Loads elements from a file given by a filename, storing them in a given array fragment.
*
* @param filename a filename.
*
 @param array an array which will be filled with data from the specified file.
* @param offset the index of the first element of {@code array} to be filled.
* @param length the number of elements of {@code array} to be filled.
* @return the number of elements actually read from the given file (it might be less than {@code length} if the file
is too short).
*/
public static int LOAD_KEYS(final CharSequence filename, final KEY_TYPE[] array, final int offset, final int
length) throws IOException {
 return LOAD_KEYS(new File(filename.toString()), array, offset, length);
}

/** Loads elements from a file given by a {@link File} object, storing them in a given array.
*
* @param file a file.
* @param array an array which will be filled with data from the specified file.
* @return the number of elements actually read from the given file (it might be less than the array length if the file
is too short).
```

```
*/
public static int LOAD_KEYS(final File file, final KEY_TYPE[] array) throws IOException
 {
 return LOAD_KEYS(file, array, 0, array.length);
}

/** Loads elements from a file given by a filename, storing them in a given array.
*
* @param filename a filename.
* @param array an array which will be filled with data from the specified file.
* @return the number of elements actually read from the given file (it might be less than the array length if the file
is too short).
*/
public static int LOAD_KEYS(final CharSequence filename, final KEY_TYPE[] array) throws IOException {
 return LOAD_KEYS(filename, array, 0, array.length);
}

/** Stores an array fragment to a given print stream.
*
* @param array an array whose elements will be written to {@code stream}.
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param stream a print stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final int offset, final int length, final PrintStream
stream) {
 PACKAGE.ARRAYS.ensureOffsetLength(array,
 offset, length);
 for(int i = 0; i < length; i++) stream.println(array[offset + i]);
}

/** Stores an array to a given print stream.
*
* @param array an array whose elements will be written to {@code stream}.
* @param stream a print stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final PrintStream stream) {
 STORE_KEYS(array, 0, array.length, stream);
}

/** Stores an array fragment to a file given by a {@link File} object.
*
* @param array an array whose elements will be written to {@code filename}.
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final int offset, final int length, final File file) throws
```

```
IOException {
 final PrintStream stream = new PrintStream(new FastBufferedOutputStream(new FileOutputStream(file)));
 STORE_KEYS(array, offset,
 length, stream);
 stream.close();
}

/** Stores an array fragment to a file given by a pathname.
*
* @param array an array whose elements will be written to {@code filename}.
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param filename a filename.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final int offset, final int length, final CharSequence
filename) throws IOException {
 STORE_KEYS(array, offset, length, new File(filename.toString()));
}

/** Stores an array to a file given by a {@link File} object.
*
* @param array an array whose elements will be written to {@code filename}.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final File file) throws IOException {
 STORE_KEYS(array, 0, array.length, file);
}

/** Stores an array to a file given by a pathname.
*
* @param array an array whose elements will be written to {@code
 filename}.
* @param filename a filename.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final CharSequence filename) throws IOException {
 STORE_KEYS(array, 0, array.length, filename);
}

/** Stores the element returned by an iterator to a given print stream.
*
* @param i an iterator whose output will be written to {@code stream}.
* @param stream a print stream.
*/
public static void STORE_KEYS(final KEY_ITERATOR i, final PrintStream stream) {
 while(i.hasNext()) stream.println(i.NEXT_KEY());
}

/** Stores the element returned by an iterator to a file given by a {@link File} object.
```

```
*
* @param i an iterator whose output will be written to {@code filename}.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_ITERATOR i, final File file) throws IOException {
 final PrintStream stream = new PrintStream(new FastBufferedOutputStream(new FileOutputStream(file)));
 STORE_KEYS(i, stream);
 stream.close();
}

/** Stores the element returned by an iterator to
 a file given by a pathname.
*
* @param i an iterator whose output will be written to {@code filename}.
* @param filename a filename.
*/
public static void STORE_KEYS(final KEY_ITERATOR i, final CharSequence filename) throws IOException {
 STORE_KEYS(i, new File(filename.toString()));
}


/** Loads elements from a given fast buffered reader, storing them in a given big-array fragment.
*
* @param reader a buffered reader.
* @param array a big array which will be filled with data from {@code reader}.
* @param offset the index of the first element of {@code array} to be filled.
* @param length the number of elements of {@code array} to be filled.
* @return the number of elements actually read from {@code reader} (it might be less than {@code length} if
{@code reader} ends).
*/
public static long LOAD_KEYS(final BufferedReader reader, final KEY_TYPE[][] array, final long offset, final
long length) throws IOException {
 ensureOffsetLength(array, offset, length);
 long c = 0;
 String
 s;
 try {
  for(int i = segment(offset); i < segment(offset + length + SEGMENT_MASK); i++) {
   final KEY_TYPE[] t = array[i];
   final int l = (int)Math.min(t.length, offset + length - start(i));
   for(int d = (int)Math.max(0, offset - start(i)); d < l; d++) {
    if ((s = reader.readLine()) != null) t[d] = KEY_CLASS.PARSE_KEY(s.trim());
    else return c;
    c++;
   }
  }
 }
 catch(EOFException itsOk) {}
```

```
 return c;
}

/** Loads elements from a given buffered reader, storing them in a given array.
*
* @param reader a buffered reader.
* @param array a big array which will be filled with data from {@code reader}.
* @return the number of elements actually read from {@code reader} (it might be less than the array length if
{@code reader} ends).
*/
public static long LOAD_KEYS(final BufferedReader reader, final KEY_TYPE[][] array) throws IOException {
 return LOAD_KEYS(reader, array, 0, length(array));
}

/** Loads elements from a file given by a {@link File} object, storing them
 in a given big-array fragment.
*
* @param file a file.
* @param array a big array which will be filled with data from the specified file.
* @param offset the index of the first element of {@code array} to be filled.
* @param length the number of elements of {@code array} to be filled.
* @return the number of elements actually read from the given file (it might be less than {@code length} if the file
is too short).
*/
public static long LOAD_KEYS(final File file, final KEY_TYPE[][] array, final long offset, final long length)
throws IOException {
 final BufferedReader reader = new BufferedReader(new FileReader(file));
 final long result = LOAD_KEYS(reader, array, offset, length);
 reader.close();

 return result;
}

/** Loads elements from a file given by a filename, storing them in a given big-array fragment.
*
* @param filename a filename.
* @param array a big array which will be filled with data from the specified file.
* @param offset the index of the first element of {@code
 array} to be filled.
* @param length the number of elements of {@code array} to be filled.
* @return the number of elements actually read from the given file (it might be less than {@code length} if the file
is too short).
*/
public static long LOAD_KEYS(final CharSequence filename, final KEY_TYPE[][] array, final long offset, final
long length) throws IOException {
 return LOAD_KEYS(new File(filename.toString()), array, offset, length);
}
```

```
/** Loads elements from a file given by a {@link File} object, storing them in a given array.
*
* @param file a file.
* @param array a big array which will be filled with data from the specified file.
* @return the number of elements actually read from the given file (it might be less than the array length if the file
is too short).
*/
public static long LOAD_KEYS(final File file, final KEY_TYPE[][] array) throws IOException {
 return LOAD_KEYS(file, array, 0, length(array));
}

/** Loads elements from a file given by a filename, storing
 them in a given array.
*
* @param filename a filename.
* @param array a big array which will be filled with data from the specified file.
* @return the number of elements actually read from the given file (it might be less than the array length if the file
is too short).
*/
public static long LOAD_KEYS(final CharSequence filename, final KEY_TYPE[][] array) throws IOException {
 return LOAD_KEYS(filename, array, 0, length(array));
}

/** Stores a big-array fragment to a given print stream.
*
* @param array a big array whose elements will be written to {@code stream}.
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param stream a print stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final long offset, final long length, final PrintStream
stream) {
 ensureOffsetLength(array, offset, length);

 for(int i = segment(offset); i < segment(offset + length + SEGMENT_MASK);
 i++) {
  final KEY_TYPE[] t = array[i];
  final int l = (int)Math.min(t.length, offset + length - start(i));
  for(int d = (int)Math.max(0, offset - start(i)); d < l; d++) stream.println(t[d]);
 }
}

/** Stores a big array to a given print stream.
*
* @param array a big array whose elements will be written to {@code stream}.
* @param stream a print stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final PrintStream stream) {
```

```
 STORE_KEYS(array, 0, length(array), stream);
}

/** Stores a big-array fragment to a file given by a {@link File} object.
*
* @param array a big array whose elements will be written to {@code filename}.
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final long offset, final long length, final File file)
throws IOException {
 final PrintStream
 stream = new PrintStream(new FastBufferedOutputStream(new FileOutputStream(file)));
 STORE_KEYS(array, offset, length, stream);
 stream.close();
}

/** Stores a big-array fragment to a file given by a pathname.
*
* @param array a big array whose elements will be written to {@code filename}.
* @param offset the index of the first element of {@code array} to be written.
* @param length the number of elements of {@code array} to be written.
* @param filename a filename.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final long offset, final long length, final
CharSequence filename) throws IOException {
 STORE_KEYS(array, offset, length, new File(filename.toString()));
}

/** Stores a big array to a file given by a {@link File} object.
*
* @param array a big array whose elements will be written to {@code filename}.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final File file) throws IOException {
 STORE_KEYS(array, 0, length(array),
 file);
}

/** Stores a big array to a file given by a pathname.
*
* @param array a big array whose elements will be written to {@code filename}.
* @param filename a filename.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final CharSequence filename) throws IOException {
 STORE_KEYS(array, 0, length(array), filename);
}
```

```
/** A wrapper that exhibits the content of a reader as a type-specific iterator. */

private static final class KEY_READER_WRAPPER implements KEY_ITERATOR {
 private final BufferedReader reader;
 private boolean toAdvance = true;
 private String s;
 private KEY_TYPE next;

 public KEY_READER_WRAPPER(final BufferedReader reader) {
  this.reader = reader;
 }

 @Override
 public boolean hasNext() {
  if (! toAdvance) return s != null;

  toAdvance = false;

  try {
   s = reader.readLine();
  }
  catch(EOFException itsOk) {}
  catch(IOException rethrow) { throw new RuntimeException(rethrow); }

  if (s == null) return false;

  next = KEY_CLASS.PARSE_KEY(s.trim());
  return
 true;
 }

 @Override
 public KEY_TYPE NEXT_KEY() {
  if (! hasNext()) throw new NoSuchElementException();
  toAdvance = true;
  return next;
 }
}



/** Wraps the given buffered reader into an iterator.
*
* @param reader a buffered reader.
*/
public static KEY_ITERATOR AS_KEY_ITERATOR(final BufferedReader reader) {
```

```
 return new KEY_READER_WRAPPER(reader);
}

/** Wraps a file given by a {@link File} object into an iterator.
*
* @param file a file.
*/
public static KEY_ITERATOR AS_KEY_ITERATOR(final File file) throws IOException {
 return new KEY_READER_WRAPPER(new BufferedReader(new FileReader(file)));
}

/** Wraps a file given by a pathname into an iterator.
*
* @param filename a filename.
*/
public static KEY_ITERATOR AS_KEY_ITERATOR(final CharSequence filename) throws IOException {
 return AS_KEY_ITERATOR(new File(filename.toString()));
}

/** Wraps a file given by a {@link File} object into an iterable object.
*
* @param file a file.
*/
public static
 KEY_ITERABLE AS_KEY_ITERABLE(final File file) {
 return () -> {
  try { return AS_KEY_ITERATOR(file); }
  catch(IOException e) { throw new RuntimeException(e); }
 };
}

/** Wraps a file given by a pathname into an iterable object.
*
* @param filename a filename.
*/
public static KEY_ITERABLE AS_KEY_ITERABLE(final CharSequence filename) {
 return () -> {
  try { return AS_KEY_ITERATOR(filename); }
  catch(IOException e) { throw new RuntimeException(e); }
 };
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/TextIOFragment.drv
No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.Iterator;
import java.util.NoSuchElementException;

/** A class providing static methods and objects that do useful things with type-specific iterators.
*
* @see Iterator
*/

public final class BIG_LIST_ITERATORS {

 private BIG_LIST_ITERATORS() {}

 /** A class returning no elements
 and a type-specific big list iterator interface.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific iterator.
  */

 public static class EmptyBigListIterator KEY_GENERIC implements KEY_BIG_LIST_ITERATOR
KEY_GENERIC, java.io.Serializable, Cloneable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected EmptyBigListIterator() {}

  @Override
  public boolean hasNext() { return false; }

  @Override
```

```
 public boolean hasPrevious() { return false; }

 @Override
 public KEY_GENERIC_TYPE NEXT_KEY() { throw new NoSuchElementException(); }

 @Override
 public KEY_GENERIC_TYPE PREV_KEY() { throw new NoSuchElementException(); }

 @Override
 public long nextIndex() { return 0; }

 @Override
 public long previousIndex() { return -1; }

 @Override
 public long skip(long n) { return 0; };

 @Override
 public long back(long n) { return 0; };

 @Override
 public Object clone() { return EMPTY_BIG_LIST_ITERATOR;
}

 private Object readResolve() { return EMPTY_BIG_LIST_ITERATOR; }
}

/** An empty iterator (immutable). It is serializable and cloneable.
 *
 * <p>The class of this objects represent an abstract empty iterator
 * that can iterate as a type-specific (list) iterator.
 */

SUPPRESS_WARNINGS_KEY_RAWTYPES
public static final EmptyBigListIterator EMPTY_BIG_LIST_ITERATOR = new EmptyBigListIterator();


/** An iterator returning a single element. */

 private static class SingletonBigListIterator KEY_GENERIC implements KEY_BIG_LIST_ITERATOR
KEY_GENERIC {
 private final KEY_GENERIC_TYPE element;
 private int curr;

 public SingletonBigListIterator(final KEY_GENERIC_TYPE element) {
  this.element = element;
 }
```

```
 @Override
 public boolean hasNext() { return curr == 0; }

 @Override
 public boolean hasPrevious() { return curr == 1; }

 @Override
 public KEY_GENERIC_TYPE NEXT_KEY() {
  if (! hasNext()) throw new NoSuchElementException();
  curr = 1;
  return element;
 }

 @Override
 public
KEY_GENERIC_TYPE PREV_KEY() {
  if (! hasPrevious()) throw new NoSuchElementException();
  curr = 0;
  return element;
 }

 @Override
 public long nextIndex() {
  return curr;
 }

 @Override
 public long previousIndex() {
  return curr - 1;
 }
}


/** Returns an iterator that iterates just over the given element.
 *
 * @param element the only element to be returned by a type-specific list iterator.
 * @return  an iterator that iterates just over {@code element}.
 */
 public static KEY_GENERIC KEY_BIG_LIST_ITERATOR KEY_GENERIC singleton(final
KEY_GENERIC_TYPE element) {
 return new SingletonBigListIterator KEY_GENERIC_DIAMOND(element);
 }


 /** An unmodifiable wrapper class for big list iterators. */

 public static class UnmodifiableBigListIterator KEY_GENERIC implements KEY_BIG_LIST_ITERATOR
KEY_GENERIC {
```

```
 protected final KEY_BIG_LIST_ITERATOR KEY_GENERIC i;

 public UnmodifiableBigListIterator(final KEY_BIG_LIST_ITERATOR KEY_GENERIC i)
{
  this.i = i;
 }

 @Override
 public boolean hasNext() { return i.hasNext(); }

 @Override
 public boolean hasPrevious() { return i.hasPrevious(); }

 @Override
 public KEY_GENERIC_TYPE NEXT_KEY() { return i.NEXT_KEY(); }

 @Override
 public KEY_GENERIC_TYPE PREV_KEY() { return i.PREV_KEY(); }

 @Override
 public long nextIndex() { return i.nextIndex(); }

 @Override
 public long previousIndex() { return i.previousIndex(); }
}

/** Returns an unmodifiable list iterator backed by the specified list iterator.
 *
 * @param i the list iterator to be wrapped in an unmodifiable list iterator.
 * @return an unmodifiable view of the specified list iterator.
 */
 public static KEY_GENERIC KEY_BIG_LIST_ITERATOR KEY_GENERIC unmodifiable(final
KEY_BIG_LIST_ITERATOR KEY_GENERIC i) { return new UnmodifiableBigListIterator
KEY_GENERIC_DIAMOND(i); }


/** A class exposing a list iterator as a big-list iterator.. */

 public static class BigListIteratorListIterator KEY_GENERIC
 implements KEY_BIG_LIST_ITERATOR KEY_GENERIC {
  protected final KEY_LIST_ITERATOR KEY_GENERIC i;

  protected BigListIteratorListIterator(final KEY_LIST_ITERATOR KEY_GENERIC i) {
   this.i = i;
  }

  private int intDisplacement(long n) {
   if (n < Integer.MIN_VALUE || n > Integer.MAX_VALUE) throw new IndexOutOfBoundsException("This big
```

```
iterator is restricted to 32-bit displacements");
  return (int)n;
 }

 @Override
 public void set(KEY_GENERIC_TYPE ok) { i.set(ok); }

 @Override
 public void add(KEY_GENERIC_TYPE ok) { i.add(ok); }

 @Override
 public int back(int n) { return i.back(n); }

 @Override
 public long back(long n) { return i.back(intDisplacement(n)); }

 @Override
 public void remove() { i.remove(); }

 @Override
 public int skip(int n) { return i.skip(n); }

 @Override
 public long skip(long n) { return i.skip(intDisplacement(n)); }

 @Override
 public boolean hasNext() { return i.hasNext(); }

 @Override
 public boolean hasPrevious() { return i.hasPrevious();
}

 @Override
 public KEY_GENERIC_TYPE NEXT_KEY() { return i.NEXT_KEY(); }

 @Override
 public KEY_GENERIC_TYPE PREV_KEY() { return i.PREV_KEY(); }

 @Override
 public long nextIndex() { return i.nextIndex(); }

 @Override
 public long previousIndex() { return i.previousIndex(); }
}

/** Returns a big-list iterator backed by the specified list iterator.
 *
 * @param i the list iterator to adapted to the big-list-iterator interface.
```

```
  * @return a big-list iterator backed by the specified list iterator.
  */
 public static KEY_GENERIC KEY_BIG_LIST_ITERATOR KEY_GENERIC asBigListIterator(final
KEY_LIST_ITERATOR KEY_GENERIC i) { return new BigListIteratorListIterator
KEY_GENERIC_DIAMOND(i); }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/BigListIterators.drv
No license file was found, but licenses were detected in source scan.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding
shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the
Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic
mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the
terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or

for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable
to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor
harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy
of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/LICENSE-2.0
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import VALUE_PACKAGE.VALUE_COLLECTION;

#if !KEY_CLASS_Object
import it.unimi.dsi.fastutil.objects.ObjectSet;
import it.unimi.dsi.fastutil.objects.ObjectIterator;
#endif

import java.util.function.Consumer;
import java.util.Map;

/** A type-specific {@link Map}; provides some additional methods that
 use polymorphism to avoid (un)boxing, and handling of a default return value.
*
* <p>Besides extending the corresponding type-specific {@linkplain it.unimi.dsi.fastutil.Function function}, this
interface strengthens {@link Map#entrySet()},
* {@link #keySet()} and {@link #values()}. Moreover, a number of methods, such as {@link #size()}, {@link
#defaultReturnValue()}, etc., are un-defaulted
* as their function default do not make sense for a map.
* Maps returning entry sets of type {@link FastEntrySet} support also fast iteration.
*
* <p>A submap or subset may or may not have an
* independent default return value (which however must be initialized to the
* default return value of the originator).
*
* @see Map
*/

public interface MAP KEY_VALUE_GENERIC extends FUNCTION KEY_VALUE_GENERIC,
Map<KEY_GENERIC_CLASS,VALUE_GENERIC_CLASS> {

 /** An entry set providing fast iteration.
  *
  * <p>In some cases (e.g., hash-based classes) iteration over an entry set requires the creation

 * of a large number of {@link java.util.Map.Entry} objects. Some {@code fastutil}
  * maps might return {@linkplain Map#entrySet() entry set} objects of type {@code FastEntrySet}: in this case,
{@link #fastIterator() fastIterator()}
  * will return an iterator that is guaranteed not to create a large number of objects, <em>possibly
  * by returning always the same entry</em> (of course, mutated), and {@link #fastForEach(Consumer)} will apply
```

```
 * the provided consumer to all elements of the entry set, <em>which might be represented
 * always by the same entry</em> (of course, mutated).
 */

interface FastEntrySet KEY_VALUE_GENERIC extends ObjectSet<MAP.Entry KEY_VALUE_GENERIC> {
 /** Returns a fast iterator over this entry set; the iterator might return always the same entry instance, suitably
mutated.
  *
  * @return a fast iterator over this entry set; the iterator might return always the same {@link java.util.Map.Entry}
instance, suitably mutated.
  */
 ObjectIterator<MAP.Entry
KEY_VALUE_GENERIC> fastIterator();

 /** Iterates quickly over this entry set; the iteration might happen always on the same entry instance, suitably
mutated.
  *
  * <p>This default implementation just delegates to {@link #forEach(Consumer)}.
  *
  * @param consumer a consumer that will by applied to the entries of this set; the entries might be represented
  * by the same entry instance, suitably mutated.
  * @since 8.1.0
  */
 default void fastForEach(final Consumer<? super MAP.Entry KEY_VALUE_GENERIC> consumer) {
  forEach(consumer);
 }
}

/**
 * Returns the number of key/value mappings in this map.  If the
 * map contains more than {@link Integer#MAX_VALUE} elements, returns {@link Integer#MAX_VALUE}.
 *
 * @return the number of key-value mappings in this map.
 * @see it.unimi.dsi.fastutil.Size64
 */

@Override
int size();

/** Removes all of the mappings from this map (optional operation).
 * The map will be empty after this call returns.
 *
 * @throws
UnsupportedOperationException if the <code>clear()</code> operation is not supported by this map
 */

@Override
default void clear() { throw new UnsupportedOperationException(); }
```

```
#if VALUES_PRIMITIVE
 /** Sets the default return value (optional operation).
  *
  * This value must be returned by type-specific versions of {@code get()},
  * {@code put()} and {@code remove()} to denote that the map does not contain
  * the specified key. It must be 0/{@code false} by default.
  *
  * @param rv the new default return value.
  * @see #defaultReturnValue()
  */
#else
 /** Sets the default return value (optional operation).
  *
  * This value must be returned by type-specific versions of {@code get()},
  * {@code put()} and {@code remove()} to denote that the map does not contain
  * the specified key. It must be {@code null} by default.
  *
  * <p><strong>Warning</strong>: Changing this to a non-null value can have
  * unforeseen consequences. Especially, it breaks compatibility with
 the
  * specifications of Java's {@link java.util.Map} interface. It has to be
  * used with great care and thorough study of all method comments is
  * recommended.
  *
  * @param rv the new default return value.
  * @see #defaultReturnValue()
  */
#endif

 @Override
 void defaultReturnValue(VALUE_GENERIC_TYPE rv);

 /** Gets the default return value.
  *
  * @return the current default return value.
  */

 @Override
 VALUE_GENERIC_TYPE defaultReturnValue();

 /** Returns a type-specific set view of the mappings contained in this map.
  *
  * <p>This method is necessary because there is no inheritance along
  * type parameters: it is thus impossible to strengthen {@link Map#entrySet()}
  * so that it returns an {@link it.unimi.dsi.fastutil.objects.ObjectSet}
  * of type-specific entries (the latter makes it possible to
  * access keys and values with type-specific methods).
```

```
  *
  * @return a type-specific set view of the mappings contained in this map.
  * @see Map#entrySet()
  */
 ObjectSet<MAP.Entry
 KEY_VALUE_GENERIC> ENTRYSET();

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
 /** Returns a set view of the mappings contained in this map.
  *  <p>Note that this specification strengthens the one given in {@link Map#entrySet()}.
  *
  * @return a set view of the mappings contained in this map.
  * @see Map#entrySet()
  * @deprecated Please use the corresponding type-specific method instead.
  */
 @Deprecated
 @Override
 @SuppressWarnings({ "unchecked", "rawtypes" })
 default ObjectSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() {
  return (ObjectSet)ENTRYSET();
 }
#else
 /** Returns a set view of the mappings contained in this map.
  *  <p>Note that this specification strengthens the one given in {@link Map#entrySet()}.
  *
  * @return a set view of the mappings contained in this map.
  * @see Map#entrySet()
  */

 @Override
 @SuppressWarnings({ "unchecked", "rawtypes" })
 default ObjectSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() {
  return
 (ObjectSet)ENTRYSET();
 }
#endif

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
 /** {@inheritDoc}
  * <p>This default implementation just delegates to the corresponding type-specific&ndash;{@linkplain
it.unimi.dsi.fastutil.Function function} method.
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
#else
 /** {@inheritDoc}
  * <p>This default implementation just delegates to the corresponding function method.
  */
#endif
```

```
 @Override
 default VALUE_GENERIC_CLASS put(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value) {
  return FUNCTION.super.put(key, value);
 }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

 /** {@inheritDoc}
  * <p>This default implementation just delegates to the corresponding type-specific&ndash;{@linkplain
it.unimi.dsi.fastutil.Function function} method.
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default VALUE_GENERIC_CLASS get(final Object key) {
  return FUNCTION.super.get(key);
 }

#endif

#if
 KEYS_PRIMITIVE || VALUES_PRIMITIVE
 /** {@inheritDoc}
  * <p>This default implementation just delegates to the corresponding type-specific&ndash;{@linkplain
it.unimi.dsi.fastutil.Function function} method.
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
#else
 /** {@inheritDoc}
  * <p>This default implementation just delegates to the corresponding type-specific&ndash;{@linkplain
it.unimi.dsi.fastutil.Function function} method.
  */
#endif
 @Override
 default VALUE_GENERIC_CLASS remove(final Object key) {
  return FUNCTION.super.remove(key);
 }

 /** {@inheritDoc}
  * <p>Note that this specification strengthens the one given in {@link Map#keySet()}.
  * @return a set view of the keys contained in this map.
  * @see Map#keySet()
  */

 @Override
 SET KEY_GENERIC keySet();

 /** {@inheritDoc}
```

```
 *  <p>Note that this specification strengthens the one given in {@link Map#values()}.
 * @return a set view of the
values contained in this map.
 * @see Map#values()
 */

@Override
VALUE_COLLECTION VALUE_GENERIC values();

#if KEYS_PRIMITIVE

/** Returns true if this function contains a mapping for the specified key.
 *
 * @param key the key.
 * @return true if this function associates a value to {@code key}.
 * @see Map#containsKey(Object)
 */

@Override
boolean containsKey(KEY_TYPE key);

/** Returns true if this function contains a mapping for the specified key.
 * <p>This default implementation just delegates to the corresponding type-specific&ndash;{@linkplain
it.unimi.dsi.fastutil.Function function} method.
 * @deprecated Please use the corresponding type-specific method instead.
 */

@Deprecated
@Override
default boolean containsKey(final Object key) {
 return FUNCTION.super.containsKey(key);
}
#else

/** Returns true if this function contains a mapping for the specified key.
 *
 * @param key the key.
 * @return true if this function associates a value to {@code key}.

* @see Map#containsKey(Object)
 */

@Override
boolean containsKey(Object key);

#endif

#if VALUES_PRIMITIVE
```

```
 /** Returns {@code true} if this map maps one or more keys to the specified value.
  * @see Map#containsValue(Object)
  */

 boolean containsValue(VALUE_TYPE value);

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */

 @Deprecated
 @Override
 default boolean containsValue(final Object value) {
  return value == null ? false : containsValue(VALUE_OBJ2TYPE(value));
 }

#endif

 // Defaultable methods

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

 /** Returns the value to which the specified key is mapped, or the {@code defaultValue} if this
  * map contains no mapping for the key.
  *
  * @param key the key.
  * @param defaultValue the default mapping of the key.
  *
  * @return the value to which the specified key is mapped, or the {@code defaultValue} if this map contains no
mapping for the key.
  *
  * @see java.util.Map#getOrDefault(Object,
 Object)
  * @since 8.0.0
  */

 default VALUE_GENERIC_TYPE getOrDefault(final KEY_TYPE key, final VALUE_GENERIC_TYPE
defaultValue) {
  final VALUE_GENERIC_TYPE v;
  return ((v = GET_VALUE(key)) != defaultReturnValue() || containsKey(key)) ? v : defaultValue;
 }

 /** If the specified key is not already associated with a value, associates it with the given
  * value and returns the {@linkplain #defaultReturnValue() default return value}, else returns
  * the current value.
  *
  * @param key key with which the specified value is to be associated.
  * @param value value to be associated with the specified key.
```

```
 *
 * @return the previous value associated with the specified key, or the {@linkplain #defaultReturnValue() default
return value} if there was no mapping for the key.
 *
 * @see java.util.Map#putIfAbsent(Object, Object)
 * @since 8.0.0
 */

default VALUE_GENERIC_TYPE putIfAbsent(final KEY_GENERIC_TYPE key, final
VALUE_GENERIC_TYPE value) {
 final VALUE_GENERIC_TYPE
v = GET_VALUE(key), drv = defaultReturnValue();
 if (v != drv || containsKey(key)) return v;
 put(key, value);
 return drv;
}

/** Removes the entry for the specified key only if it is currently mapped to the specified value.
 *
 * @param key key with which the specified value is associated.
 * @param value value expected to be associated with the specified key.
 *
 * @return {@code true} if the value was removed.
 *
 * @see java.util.Map#remove(Object, Object)
 * @since 8.0.0
 */

default boolean remove(final KEY_TYPE key, final VALUE_TYPE value) {
 final VALUE_GENERIC_TYPE curValue = GET_VALUE(key);
 if (!VALUE_EQUALS(curValue, value) || (curValue == defaultReturnValue() && !containsKey(key))) return false;
 REMOVE_VALUE(key);
 return true;
}

/** Replaces the entry for the specified key only if currently mapped to the specified value.
 *
 * @param key key with which the specified value is associated.
 * @param oldValue value expected to be associated
with the specified key.
 * @param newValue value to be associated with the specified key.
 *
 * @return {@code true} if the value was replaced.
   *
 * @see java.util.Map#replace(Object, Object, Object)
 * @since 8.0.0
 */
```

```
 default boolean replace(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE oldValue, final
VALUE_GENERIC_TYPE newValue) {
  final VALUE_GENERIC_TYPE curValue = GET_VALUE(key);
  if (!VALUE_EQUALS(curValue, oldValue) || (curValue == defaultReturnValue() && !containsKey(key))) return
false;
  put(key, newValue);
  return true;
 }

 /** Replaces the entry for the specified key only if it is currently mapped to some value.
  *
  * @param key key with which the specified value is associated.
  * @param value value to be associated with the specified key.
  *
  * @return the previous value associated with the specified key, or the {@linkplain #defaultReturnValue() default
return value} if there was no mapping for the key.
  *
  * @see java.util.Map#replace(Object,
 Object)
  * @since 8.0.0
  */

 default VALUE_GENERIC_TYPE replace(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value) { return containsKey(key) ? put(key, value) : defaultReturnValue(); }

#ifdef JDK_PRIMITIVE_FUNCTION

 /** If the specified key is not already associated with a value, attempts to compute its value
  * using the given mapping function and enters it into this map.
  *
  * <p>Note that contrarily to the default {@linkplain java.util.Map#computeIfAbsent(Object,
java.util.function.Function) computeIfAbsent()},
  * it is not possible to not add a value for a given key, since the {@code mappingFunction} cannot
  * return {@code null}. If such a behavior is needed, please use the corresponding <em>nullable</em> version.
  *
  * @param key key with which the specified value is to be associated.
  * @param mappingFunction the function to compute a value.
  *
  * @return the current (existing or computed) value associated with the specified key.
  *
  * @see java.util.Map#computeIfAbsent(Object,
 java.util.function.Function)
  * @since 8.0.0
  */

 default VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_JDK(final KEY_GENERIC_TYPE key, final
JDK_PRIMITIVE_FUNCTION KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) {
  java.util.Objects.requireNonNull(mappingFunction);
```

```
  final VALUE_GENERIC_TYPE v = GET_VALUE(key);
  if (v != defaultReturnValue() || containsKey(key)) return v;

  VALUE_GENERIC_TYPE newValue =
VALUE_NARROWING(mappingFunction.JDK_PRIMITIVE_FUNCTION_APPLY(key));
  put(key, newValue);
  return newValue;
 }

#endif

#if KEYS_PRIMITIVE && VALUES_PRIMITIVE

 /** If the specified key is not already associated with a value, attempts to compute its value
  * using the given mapping function and enters it into this map unless it is {@code null}.
  *
  * <p>Note that this version of {@linkplain java.util.Map#computeIfAbsent(Object, java.util.function.Function)
computeIfAbsent()}
  * should be used only if you plan to return {@code null} in the mapping
 function.
  *
  * @param key key with which the specified value is to be associated.
  * @param mappingFunction the function to compute a value.
  *
  * @return the current (existing or computed) value associated with the specified key,
  * or the {@linkplain #defaultReturnValue() default return value} if the computed value is {@code null}.
  *
  * @see java.util.Map#computeIfAbsent(Object, java.util.function.Function)
  * @since 8.0.0
  */

 default VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_NULLABLE(final KEY_GENERIC_TYPE key,
final JDK_KEY_TO_GENERIC_FUNCTION<? extends VALUE_GENERIC_CLASS> mappingFunction) {
  java.util.Objects.requireNonNull(mappingFunction);
  final VALUE_GENERIC_TYPE v = GET_VALUE(key), drv = defaultReturnValue();
  if (v != drv || containsKey(key)) return v;

  VALUE_GENERIC_CLASS mappedValue = mappingFunction.apply(key);
  if (mappedValue == null) return drv;
  VALUE_GENERIC_TYPE newValue = VALUE_CLASS2TYPE(mappedValue);
  put(key, newValue);
  return newValue;
 }

#endif

 /**
```

```
 If the specified key is not already associated with a value, attempts to compute its value
  * using the given mapping function and enters it into this map, unless the key is not present
  * in the given mapping function.
  *
  * <p>This version of {@linkplain java.util.Map#computeIfAbsent(Object, java.util.function.Function)
computeIfAbsent()}
  * uses a type-specific version of <code>fastutil</code>'s {@link it.unimi.dsi.fastutil.Function Function}.
  * Since {@link it.unimi.dsi.fastutil.Function Function} has a {@link
it.unimi.dsi.fastutil.Function#containsKey(Object) containsKey()}
  * method, it is possible to avoid adding a key by having {@code containsKey()} return {@code false} for that key.
  *
  * @param key key with which the specified value is to be associated.
  * @param mappingFunction the function to compute a value.
  *
  * @return the current (existing or computed) value associated with the specified key.
  *
  * @see java.util.Map#computeIfAbsent(Object,
 java.util.function.Function)
  * @since 8.0.0
  */

 default VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_PARTIAL(final KEY_GENERIC_TYPE key, final
FUNCTION KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) {
  java.util.Objects.requireNonNull(mappingFunction);
  final VALUE_GENERIC_TYPE v = GET_VALUE(key), drv = defaultReturnValue();
  if (v != drv || containsKey(key)) return v;

  if (!mappingFunction.containsKey(key)) return drv;
  VALUE_GENERIC_TYPE newValue = mappingFunction.GET_VALUE(key);
  put(key, newValue);
  return newValue;
 }

 /** If the value for the specified key is present, attempts to compute a new mapping given the key and its current
mapped value.
  *
  * @param key key with which the specified value is to be associated.
  * @param remappingFunction the function to compute a value.
  *
  * @return the new value associated with the specified key, or the {@linkplain #defaultReturnValue() default return
value} if none.
  *
  * @see java.util.Map#computeIfPresent(Object,
 java.util.function.BiFunction)
  * @since 8.0.0
  */

 default VALUE_GENERIC_TYPE COMPUTE_IF_PRESENT(final KEY_GENERIC_TYPE key, final
```

```
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) {
  java.util.Objects.requireNonNull(remappingFunction);

  final VALUE_GENERIC_TYPE oldValue = GET_VALUE(key), drv = defaultReturnValue();
  if (oldValue == drv && !containsKey(key)) return drv;
  final VALUE_GENERIC_CLASS newValue = remappingFunction.apply(KEY2OBJ(key),
VALUE2OBJ(oldValue));

  if (newValue == null) { REMOVE_VALUE(key); return drv; }

#if VALUES_PRIMITIVE
  VALUE_GENERIC_TYPE newVal = VALUE_CLASS2TYPE(newValue);
  put(key, newVal);
  return newVal;
#else
  put(key, newValue);
  return newValue;
#endif
 }

 /** Attempts to compute a mapping for the specified key and its current mapped value (or {@code null} if there is
no current mapping).
  *
  * <p>If the function returns {@code null}, the mapping
 is removed (or remains absent if initially absent).
  * If the function itself throws an (unchecked) exception, the exception is rethrown, and the current mapping is left
unchanged.
  *
  * @param key key with which the specified value is to be associated.
  * @param remappingFunction the function to compute a value.
  *
  * @return the new value associated with the specified key, or the {@linkplain #defaultReturnValue() default return
value} if none.
  *
  * @see java.util.Map#compute(Object, java.util.function.BiFunction)
  * @since 8.0.0
  */

 default VALUE_GENERIC_TYPE COMPUTE(final KEY_GENERIC_TYPE key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) {
  java.util.Objects.requireNonNull(remappingFunction);
  final VALUE_GENERIC_TYPE oldValue = GET_VALUE(key), drv = defaultReturnValue();
  final boolean contained = oldValue != drv || containsKey(key);
  final VALUE_GENERIC_CLASS newValue
 = remappingFunction.apply(KEY2OBJ(key), contained ? VALUE2OBJ(oldValue) : null);
```

```
  if (newValue == null) {
   if (contained) REMOVE_VALUE(key);
   return drv;
  }
#if VALUES_PRIMITIVE
  final VALUE_GENERIC_TYPE newVal = VALUE_CLASS2TYPE(newValue);
  put(key, newVal);
  return newVal;
#else
  put(key, newValue);
  return newValue;
#endif
 }

#if VALUES_PRIMITIVE
 /**
  * If the specified key is not already associated with a value, associates it with the given {@code value}.
  * Otherwise, replaces the associated value with the results of the given remapping function, or removes if the result
is {@code null}.
  *
  * @param key key with which the resulting value is to be associated.
  * @param value the value to be merged with the existing value associated with the key or, if no existing value is
associated with the key, to be associated with the key.
  * @param remappingFunction the function to recompute a value if present.
  *
  * @return the new value associated with the specified key,
 or the {@linkplain #defaultReturnValue() default return value} if no value is associated with the key.
  *
  * @see java.util.Map#merge(Object, Object, java.util.function.BiFunction)
  * @since 8.0.0
  */
#else
 /**
  * If the specified key is not already associated with a value or is associated with null, associates it with the given
non-null {@code value}.
  * Otherwise, replaces the associated value with the results of the given remapping function, or removes if the result
is {@code null}.
  *
  * @param key key with which the resulting value is to be associated.
  * @param value the non-{@code null} value to be merged with the existing value associated with the key or, if no
existing value is associated with the key, to be associated with the key.
  * @param remappingFunction the function to recompute a value if present.
  *
  * @return the new value associated with the specified key, or the {@linkplain #defaultReturnValue() default return
value} if no value is associated with
 the key.
  *
  * @see java.util.Map#merge(Object, Object, java.util.function.BiFunction)
```

```
  * @since 8.0.0
  */
#endif

 default VALUE_GENERIC_TYPE MERGE(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value, final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) {
  java.util.Objects.requireNonNull(remappingFunction);
#if VALUES_REFERENCE
  java.util.Objects.requireNonNull(value);
#endif
  final VALUE_GENERIC_TYPE oldValue = GET_VALUE(key), drv = defaultReturnValue();
  final VALUE_GENERIC_TYPE newValue;
  if (oldValue != drv || containsKey(key)) {
   final VALUE_GENERIC_CLASS mergedValue = remappingFunction.apply(VALUE2OBJ(oldValue),
VALUE2OBJ(value));
   if (mergedValue == null) { REMOVE_VALUE(key); return drv; }
   newValue = VALUE_CLASS2TYPE(mergedValue);
  } else {
   newValue = value;
  }

  put(key, newValue);
  return newValue;
 }

 /** {@inheritDoc}
  * <p>This default implementation
 just delegates to the corresponding {@link Map} method.
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default VALUE_GENERIC_CLASS getOrDefault(final Object key, final VALUE_GENERIC_CLASS
defaultValue) {
  return Map.super.getOrDefault(key, defaultValue);
 }

 /** {@inheritDoc}
  * <p>This default implementation just delegates to the corresponding {@link Map} method.
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default VALUE_GENERIC_CLASS putIfAbsent(final KEY_GENERIC_CLASS key, final
VALUE_GENERIC_CLASS value) {
  return Map.super.putIfAbsent(key, value);
 }

 /** {@inheritDoc}
```

```
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default boolean remove(final Object key, final Object value) {
 return Map.super.remove(key, value);
}

 /** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default boolean replace(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS oldValue, final
VALUE_GENERIC_CLASS newValue) {
 return Map.super.replace(key, oldValue, newValue);
}

/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS replace(final KEY_GENERIC_CLASS key, final
VALUE_GENERIC_CLASS value) {
 return Map.super.replace(key, value);
}

#if KEYS_PRIMITIVE

/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default
VALUE_GENERIC_CLASS computeIfAbsent(final KEY_GENERIC_CLASS key, final
java.util.function.Function<? super KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS>
mappingFunction) {
 return Map.super.computeIfAbsent(key, mappingFunction);
}

/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS computeIfPresent(final KEY_GENERIC_CLASS key, final
```

```
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) {
  return Map.super.computeIfPresent(key, remappingFunction);
 }

 /** {@inheritDoc}
  * <p>This default implementation just delegates to the corresponding {@link Map} method.
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default VALUE_GENERIC_CLASS compute(final KEY_GENERIC_CLASS
 key, final java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ?
extends VALUE_GENERIC_CLASS> remappingFunction) {
  return Map.super.compute(key, remappingFunction);
 }

#endif

 /** {@inheritDoc}
  * <p>This default implementation just delegates to the corresponding {@link Map} method.
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default VALUE_GENERIC_CLASS merge(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value, final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) {
  return Map.super.merge(key, value, remappingFunction);
 }

#endif

 /** A type-specific {@link java.util.Map.Entry}; provides some additional methods
  *  that use polymorphism to avoid (un)boxing.
  *
  * @see java.util.Map.Entry
  */

 interface Entry KEY_VALUE_GENERIC extends Map.Entry
<KEY_GENERIC_CLASS,VALUE_GENERIC_CLASS>
 {

#if KEYS_PRIMITIVE
  /** Returns the key corresponding to this entry.
   * @see java.util.Map.Entry#getKey()
   */
  KEY_TYPE ENTRY_GET_KEY();

  /** {@inheritDoc}
```

```
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  default KEY_GENERIC_CLASS getKey() {
   return KEY2OBJ(ENTRY_GET_KEY());
  }
#endif

#if VALUES_PRIMITIVE
  /** Returns the value corresponding to this entry.
   * @see java.util.Map.Entry#getValue()
   */
  VALUE_TYPE ENTRY_GET_VALUE();

  /** Replaces the value corresponding to this entry with the specified value (optional operation).
   * @see java.util.Map.Entry#setValue(Object)
   */
  VALUE_TYPE setValue(final VALUE_TYPE value);

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  default VALUE_GENERIC_CLASS getValue() {
   return VALUE2OBJ(ENTRY_GET_VALUE());
  }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding
 type-specific method instead. */
  @Deprecated
  @Override
  default VALUE_GENERIC_CLASS setValue(final VALUE_GENERIC_CLASS value) {
   return VALUE2OBJ(setValue(VALUE_CLASS2TYPE(value)));
  }
#endif
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Map.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

#if KEY_CLASS_Object
import java.util.Comparator;
#endif

import java.util.Arrays;

/** A class providing static methods and objects that do useful things with indirect heaps.
*
* <p>An indirect heap is an extension of a semi-indirect heap using also an
* <em>inversion array</em> of
 the same length as the reference array,
* satisfying the relation {@code heap[inv[i]]==i} when
* {@code inv[i]&gt;=0}, and {@code inv[heap[i]]==i} for all elements in the heap.
*/

public final class INDIRECT_HEAPS {

 private INDIRECT_HEAPS() {}

 /** Moves the given element down into the indirect heap until it reaches the lowest possible position.
  *
  * @param refArray the reference array.
  * @param heap the indirect heap (starting at 0).
  * @param inv the inversion array.
  * @param size the number of elements in the heap.
  * @param i the index in the heap of the element to be moved down.
  * @param c a type-specific comparator, or {@code null} for the natural order.
  * @return the new position in the heap of the element of heap index {@code i}.
  */

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public static KEY_GENERIC int downHeap(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int[]
inv, final int size, int i, final KEY_COMPARATOR KEY_GENERIC c) {
```

```
 assert i < size;

 final
int e = heap[i];
 final KEY_GENERIC_TYPE E = refArray[e];
 int child;

 if (c == null)
  while ((child = (i << 1) + 1) < size) {
   int t = heap[child];
   final int right = child + 1;
   if (right < size && KEY_LESS(refArray[heap[right]], refArray[t])) t = heap[child = right];
   if (KEY_LESSEQ(E, refArray[t])) break;
   heap[i] = t;
   inv[heap[i]] = i;
   i = child;
  }
 else
  while ((child = (i << 1) + 1) < size) {
   int t = heap[child];
   final int right = child + 1;
   if (right < size && c.compare(refArray[heap[right]], refArray[t]) < 0) t = heap[child = right];
   if (c.compare(E, refArray[t]) <= 0) break;
   heap[i] = t;
   inv[heap[i]] = i;
   i = child;
  }

 heap[i] = e;
 inv[e] = i;
 return i;
}

/** Moves the given element up in the indirect heap until it reaches the highest possible position.
 *
 * Note that in principle after this call the heap property may be violated.
 *
 * @param refArray the reference array.
 * @param heap
the indirect heap (starting at 0).
 * @param inv the inversion array.
 * @param size the number of elements in the heap.
 * @param i the index in the heap of the element to be moved up.
 * @param c a type-specific comparator, or {@code null} for the natural order.
 * @return the new position in the heap of the element of heap index {@code i}.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
```

```
 public static KEY_GENERIC int upHeap(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int[] inv,
final int size, int i, final KEY_COMPARATOR KEY_GENERIC c) {
  assert i < size;

  final int e = heap[i];
  final KEY_GENERIC_TYPE E = refArray[e];

  if (c == null)
   while (i != 0) {
    final int parent = (i - 1) >>> 1;
    final int t = heap[parent];
    if (KEY_LESSEQ(refArray[t], E)) break;
    heap[i] = t;
    inv[heap[i]] = i;
    i = parent;
   }
  else
   while (i != 0) {
    final int parent = (i - 1) >>> 1;
    final int t = heap[parent];
    if (c.compare(refArray[t], E) <= 0) break;
    heap[i]
= t;
    inv[heap[i]] = i;
    i = parent;
   }

  heap[i] = e;
  inv[e] = i;

  return i;
 }

 /** Creates an indirect heap in the given array.
  *
  * @param refArray the reference array.
  * @param offset the first element of the reference array to be put in the heap.
  * @param length the number of elements to be put in the heap.
  * @param heap the array where the heap is to be created.
  * @param inv the inversion array.
  * @param c a type-specific comparator, or {@code null} for the natural order.
  */

 public static KEY_GENERIC void makeHeap(final KEY_GENERIC_TYPE[] refArray, final int offset, final int
length, final int[] heap, final int[] inv, final KEY_COMPARATOR KEY_GENERIC c) {
  ARRAYS.ensureOffsetLength(refArray, offset, length);
  if (heap.length < length) throw new IllegalArgumentException("The heap length (" + heap.length + ") is smaller
than the number of elements (" + length + ")");
```

```
  if (inv.length < refArray.length) throw new IllegalArgumentException("The
 inversion array length (" + heap.length + ") is smaller than the length of the reference array (" + refArray.length +
")");

  Arrays.fill(inv, 0, refArray.length, -1);

  int i = length;
  while(i-- != 0) inv[heap[i] = offset + i] = i;

  i = length >>> 1;
  while(i-- != 0) downHeap(refArray, heap, inv, length, i, c);
 }


 /** Creates an indirect heap from a given index array.
  *
  * @param refArray the reference array.
  * @param heap an array containing indices into {@code refArray}.
  * @param inv the inversion array.
  * @param size the number of elements in the heap.
  * @param c a type-specific comparator, or {@code null} for the natural order.
  */

 public static KEY_GENERIC void makeHeap(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int[]
inv, final int size, final KEY_COMPARATOR KEY_GENERIC c) {
  int i = size >>> 1;
  while(i-- != 0) downHeap(refArray, heap, inv, size, i, c);
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/IndirectHeaps.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
*/


package PACKAGE;

#if KEY_CLASS_Object
import java.util.Comparator;
import it.unimi.dsi.fastutil.IndirectPriorityQueue;
#endif

import java.util.NoSuchElementException;

import it.unimi.dsi.fastutil.ints.IntArrays;

/** A type-specific heap-based semi-indirect priority queue.
*
* <p>Instances of this
 class use as reference list a <em>reference array</em>,
* which must be provided to each constructor. The priority queue is
* represented using a heap. The heap is enlarged as needed, but it is never
* shrunk. Use the {@link #trim()} method to reduce its size, if necessary.
*
* <p>This implementation allows one to enqueue several time the same index, but
* you must be careful when calling {@link #changed()}.
*/

public class HEAP_SEMI_INDIRECT_PRIORITY_QUEUE KEY_GENERIC implements
INDIRECT_PRIORITY_QUEUE KEY_GENERIC {

 /** The reference array. */
 protected final KEY_GENERIC_TYPE refArray[];

 /** The semi-indirect heap. */
 protected int heap[] = IntArrays.EMPTY_ARRAY;

 /** The number of elements in this queue. */
 protected int size;

 /** The type-specific comparator used in this queue. */
 protected KEY_COMPARATOR KEY_SUPER_GENERIC c;

 /** Creates a new empty queue without elements with a given capacity and comparator.
  *
  * @param refArray the reference array.
  * @param
 capacity the initial capacity of this queue.
  * @param c the comparator used in this queue, or {@code null} for the natural order.
```

```
 */
 public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, int capacity,
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
  if (capacity > 0) this.heap = new int[capacity];
  this.refArray = refArray;
  this.c = c;
 }

 /** Creates a new empty queue with given capacity and using the natural order.
  *
  * @param refArray the reference array.
  * @param capacity the initial capacity of this queue.
  */
 public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, int capacity) {
  this(refArray, capacity, null);
 }

 /** Creates a new empty queue with capacity equal to the length of the reference array and a given comparator.
  *
  * @param refArray the reference array.
  * @param c the comparator used in this queue, or {@code null} for the natural order.
  */
 public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray,
KEY_COMPARATOR
 KEY_SUPER_GENERIC c) {
  this(refArray, refArray.length, c);
 }

 /** Creates a new empty queue with capacity equal to the length of the reference array and using the natural order.
  * @param refArray the reference array.
  */
 public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray) {
  this(refArray, refArray.length, null);
 }


 /** Wraps a given array in a queue using a given comparator.
  *
  * <p>The queue returned by this method will be backed by the given array.
  * The first {@code size} element of the array will be rearranged so to form a heap (this is
  * more efficient than enqueing the elements of {@code a} one by one).
  *
  * @param refArray the reference array.
  * @param a an array of indices into {@code refArray}.
  * @param size the number of elements to be included in the queue.
  * @param c the comparator used in this queue, or {@code null} for the natural order.
  */
 public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray,
```

```
 final int[] a, int size, final KEY_COMPARATOR KEY_SUPER_GENERIC c) {
  this(refArray, 0, c);
  this.heap = a;
  this.size = size;
  SEMI_INDIRECT_HEAPS.makeHeap(refArray, a, size, c);
 }


 /** Wraps a given array in a queue using a given comparator.
  *
  * <p>The queue returned by this method will be backed by the given array.
  * The elements of the array will be rearranged so to form a heap (this is
  * more efficient than enqueing the elements of {@code a} one by one).
  *
  * @param refArray the reference array.
  * @param a an array of indices into {@code refArray}.
  * @param c the comparator used in this queue, or {@code null} for the natural order.
  */
 public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, final
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
  this(refArray, a, a.length, c);
 }

 /** Wraps a given array in a queue using the natural order.
  *
  * <p>The queue returned by this method will be backed by the given array.
  * The
 first {@code size} element of the array will be rearranged so to form a heap (this is
  * more efficient than enqueing the elements of {@code a} one by one).
  *
  * @param refArray the reference array.
  * @param a an array of indices into {@code refArray}.
  * @param size the number of elements to be included in the queue.
  */
 public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, int
size) {
  this(refArray, a, size, null);
 }


 /** Wraps a given array in a queue using the natural order.
  *
  * <p>The queue returned by this method will be backed by the given array.
  * The elements of the array will be rearranged so to form a heap (this is
  * more efficient than enqueing the elements of {@code a} one by one).
  *
  * @param refArray the reference array.
  * @param a an array of indices into {@code refArray}.
```

```
 */
public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a) {
 this(refArray, a, a.length);
}

/**
Ensures that the given index is a valid reference.
 *
 * @param index an index in the reference array.
 * @throws IndexOutOfBoundsException if the given index is negative or larger than the reference array length.
 */
protected void ensureElement(final int index) {
 if (index < 0)  throw new IndexOutOfBoundsException("Index (" + index + ") is negative");
 if (index >= refArray.length) throw new IndexOutOfBoundsException("Index (" + index + ") is larger than or equal
to reference array size (" + refArray.length + ")");
}

@Override
public void enqueue(int x) {
 ensureElement(x);

 if (size == heap.length) heap = IntArrays.grow(heap, size + 1);

 heap[size++] = x;
 SEMI_INDIRECT_HEAPS.upHeap(refArray, heap, size, size - 1, c);
}

@Override
public int dequeue() {
 if (size == 0) throw new NoSuchElementException();
 final int result = heap[0];
 heap[0] = heap[--size];
 if (size != 0) SEMI_INDIRECT_HEAPS.downHeap(refArray, heap, size, 0, c);
 return result;
}

@Override
public
int first() {
 if (size == 0) throw new NoSuchElementException();
 return heap[0];
}

/** {@inheritDoc}
 *
 * <p>The caller <strong>must</strong> guarantee that when this method is called the
 * index of the first element appears just once in the queue. Failure to do so
 * will bring the queue in an inconsistent state, and will cause
```

```
 * unpredictable behaviour.
 */
@Override
public void changed() {
 SEMI_INDIRECT_HEAPS.downHeap(refArray, heap, size, 0, c);
}

/** Rebuilds this heap in a bottom-up fashion (in linear time). */

@Override
public void allChanged() { SEMI_INDIRECT_HEAPS.makeHeap(refArray, heap, size, c); }

@Override
public int size() { return size; }

@Override
public void clear() { size = 0; }

/** Trims the backing array so that it has exactly {@link #size()} elements. */

public void trim() {
 heap = IntArrays.trim(heap, size);
}

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return c; }

/** Writes in the provided
array the <em>front</em> of the queue, that is, the set of indices
 * whose elements have the same priority as the top.
 * @param a an array whose initial part will be filled with the frnot (must be sized as least as the heap size).
 * @return the number of elements of the front.
 */
@Override
public int front(final int[] a) { return c == null ? SEMI_INDIRECT_HEAPS.front(refArray, heap, size, a) :
SEMI_INDIRECT_HEAPS.front(refArray, heap, size, a, c); }

@Override
public String toString() {
 StringBuffer s = new StringBuffer();
 s.append("[");
 for (int i = 0; i < size; i++) {
  if (i != 0) s.append(", ");
  s.append(refArray[heap [i]]);
 }
 s.append("]");
 return s.toString();
}
```

```
#ifdef TEST

 /** The original class, now just used for testing. */

 private static class TestQueue {

  /** The reference array */
  private KEY_TYPE refArray[];
  /** Its length */
  private int N;
  /** The number of elements in the heaps */
  private int n;
  /** The two comparators
 */
  private KEY_COMPARATOR primaryComp, secondaryComp;
  /** Two indirect heaps are used, called {@code primary} and {@code secondary}. Each of them contains
   a permutation of {@code n} among the indices 0, 1, ..., {@code N}-1 in such a way that the corresponding
   objects be sorted with respect to the two comparators.
   We also need an array {@code inSec[]} so that {@code inSec[k]} is the index of {@code secondary}
   containing {@code k}.
  */
  private int primary[], secondary[], inSec[];

  /** Builds a double indirect priority queue.
   *  @param refArray The reference array.
   *  @param primaryComp The primary comparator.
   *  @param secondaryComp The secondary comparator.
   */
  public TestQueue(KEY_TYPE refArray[], KEY_COMPARATOR primaryComp, KEY_COMPARATOR
secondaryComp) {
   this.refArray = refArray;
   this.N = refArray.length;
   assert this.N != 0;
   this.n = 0;
   this.primaryComp = primaryComp;
   this.secondaryComp = secondaryComp;
   this.primary = new int[N];
   this.secondary
 = new int[N];
   this.inSec = new int[N];
   java.util.Arrays.fill(inSec, -1);
  }

  /** Adds an index to the queue. Notice that the index should not be already present in the queue.
   *  @param i The index to be added
   */
  public void add(int i) {
```

```
  if (i < 0 || i >= refArray.length) throw new IndexOutOfBoundsException();
  //if (inSec[i] >= 0) throw new IllegalArgumentException();
  primary[n] = i;
  n++;
  swimPrimary(n-1);
 }

 /** Heapify the primary heap.
  *  @param i The index of the heap to be heapified.
  */
 private void heapifyPrimary(int i) {
  int dep = primary[i];
  int child;

  while ((child = 2*i+1) < n) {
   if (child+1 < n && primaryComp.compare(refArray[primary[child+1]], refArray[primary[child]]) < 0) child++;
   if (primaryComp.compare(refArray[dep], refArray[primary[child]]) <= 0) break;
   primary[i] = primary[child];
   i = child;
  }
  primary[i] = dep;
 }

 /** Heapify the secondary heap.
  *  @param i The index
of the heap to be heapified.
  */
 private void heapifySecondary(int i) {
  int dep = secondary[i];
  int child;

  while ((child = 2*i+1) < n) {
   if (child+1 < n && secondaryComp.compare(refArray[secondary[child+1]], refArray[secondary[child]]) < 0)
child++;
   if (secondaryComp.compare(refArray[dep], refArray[secondary[child]]) <= 0) break;
   secondary[i] = secondary[child]; inSec[secondary[i]] = i;
   i = child;
  }
  secondary[i] = dep; inSec[secondary[i]] = i;
 }

 /** Swim and heapify the primary heap.
  *  @param i The index to be moved.
  */
 private void swimPrimary(int i) {
  int dep = primary[i];
  int parent;
```

```
 while (i != 0 && (parent = (i - 1) / 2) >= 0) {
  if (primaryComp.compare(refArray[primary[parent]], refArray[dep]) <= 0) break;
  primary[i] = primary[parent];
  i = parent;
 }
 primary[i] = dep;
 heapifyPrimary(i);
}

/** Swim and heapify the secondary heap.
 *  @param i The index to be moved.
 */
private void swimSecondary(int
i) {
 int dep = secondary[i];
 int parent;

 while (i != 0 && (parent = (i - 1) / 2) >= 0) {
  if (secondaryComp.compare(refArray[secondary[parent]], refArray[dep]) <= 0) break;
  secondary[i] = secondary[parent]; inSec[secondary[i]] = i;
  i = parent;
 }
 secondary[i] = dep; inSec[secondary[i]] = i;
 heapifySecondary(i);
}

/** Returns the minimum element with respect to the primary comparator.
 @return the minimum element.
*/
public int top() {
 if (n == 0) throw new NoSuchElementException();
 return primary[0];
}

/** Returns the minimum element with respect to the secondary comparator.
 @return the minimum element.
*/
public int secTop() {
 if (n == 0) throw new NoSuchElementException();
 return secondary[0];
}

/** Removes the minimum element with respect to the primary comparator.
 *  @return the removed element.
 */
public void remove() {
 if (n == 0) throw new NoSuchElementException();
 int result = primary[0];
```

```
  //
 Copy a leaf
  primary[0] = primary[n-1];
  n--;
  heapifyPrimary(0);
  return;
 }

 public void clear() {
  while(size() != 0) remove();
 }

 /** Signals that the minimum element with respect to the comparator has changed.
  */
 public void change() {
  heapifyPrimary(0);
 }

 /** Returns the number of elements in the queue.
  *  @return the size of the queue
  */
 public int size() {
  return n;
 }

 public String toString() {
  String s = "[";
  for (int i = 0; i < n; i++)
   s += refArray[primary[i]]+", ";
  return s+ "]";
 }
}

 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif KEY_CLASS_Object
  return Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }
```

```
 private
 static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
 private static java.text.FieldPosition p = new java.text.FieldPosition(0);

 private static String format(double d) {
  StringBuffer s = new StringBuffer();
  return format.format(d, s, p).toString();
 }

 private static void speedTest(int n, boolean comp) {
  System.out.println("There are presently no speed tests for this class.");
 }


 private static void fatal(String msg) {
  System.out.println(msg);
  System.exit(1);
 }

 private static void ensure(boolean cond, String msg) {
  if (cond) return;
  fatal(msg);
 }

 private static boolean heapEqual(int[] a, int[] b, int sizea, int sizeb) {
  if (sizea != sizeb) return false;
  while(sizea-- != 0) if (a[sizea] != b[sizea]) return false;
  return true;
 }

 protected static void runTest(int n) {
  long ms;
  Exception mThrowsIllegal, tThrowsIllegal, mThrowsOutOfBounds, tThrowsOutOfBounds, mThrowsNoElement,
tThrowsNoElement;
  int rm = 0, rt = 0;
  KEY_TYPE[]
 refArray = new KEY_TYPE[n];

  for(int i = 0; i < n; i++) refArray[i] = genKey();

  HEAP_SEMI_INDIRECT_PRIORITY_QUEUE m = new
HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(refArray, COMPARATORS.NATURAL_COMPARATOR);
  TestQueue t = new TestQueue(refArray, COMPARATORS.NATURAL_COMPARATOR,
COMPARATORS.OPPOSITE_COMPARATOR);

  /* We add pairs to t. */
  for(int i = 0; i < n / 2;  i++) {
   t.add(i);
```

```
  m.enqueue(i);
 }

 ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after creation (" + m + ",
" + t + ")");

 /* Now we add and remove random data in m and t, checking that the result is the same. */

 for(int i=0; i<2*n;  i++) {
  if (r.nextDouble() < 0.01) {
   t.clear();
   m.clear();
   for(int j = 0; j < n / 2;  j++) {
    t.add(j);
    m.enqueue(j);
   }
  }

  int T = r.nextInt(2 * n);

  mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

  try {
   m.enqueue(T);
  }
  catch
 (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
  catch (IllegalArgumentException e) { mThrowsIllegal = e; }

  try {
   t.add(T);
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
  catch (IllegalArgumentException e) { tThrowsIllegal = e; }

  ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): enqueue()
divergence in IndexOutOfBoundsException for " + T + " (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds
+ ")");
  ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): enqueue() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");

  ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after enqueue (" + m +
", " + t + ")");

  if (m.size() != 0) {
   ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after enqueue (" + m.first()
```

```
 + ", " + t.top() + ")");
   }

   mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

   try {
    rm = m.dequeue();
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { mThrowsIllegal = e; }
   catch (NoSuchElementException e) { mThrowsNoElement = e; }

   try {
    rt = t.top();
    t.remove();
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { tThrowsIllegal = e; }
   catch (NoSuchElementException e) { tThrowsNoElement = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): dequeue()
divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");
   ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): dequeue() divergence in
IllegalArgumentException  (" + mThrowsIllegal + ", " + tThrowsIllegal
 + ")");
   ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): dequeue() divergence
in NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
   if (mThrowsOutOfBounds == null) ensure(rt == rm , "Error (" + seed + "): divergence in dequeue() between t and m
(" + rt + ", " + rm + ")");


   ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after dequeue (" + m +
", " + t + ")");

   if (m.size() != 0) {
    ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after dequeue (" + m.first() + ", " +
t.top() + ")");
   }

   if (m.size() != 0 && ((new it.unimi.dsi.fastutil.ints.IntOpenHashSet(m.heap, 0, m.size)).size() == m.size())) {

    refArray[m.first()] = genKey();

    m.changed();
    t.change();

    ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after change (" + m + ",
```

```
" + t + ")");

    if
 (m.size() != 0) {
     ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after change (" + m.first() + ", " +
t.top() + ")");
    }
   }
  }


  /* Now we check that m actually holds the same data. */

  m.clear();
  ensure(m.isEmpty(), "Error (" + seed + "): m is not empty after clear()");

  System.out.println("Test OK");
 }



 public static void main(String args[]) {
  int n  = Integer.parseInt(args[1]);
  if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));


  try {
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
   else if ("test".equals(args[0])) runTest(n);
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed: " + seed);
  }
 }

#endif
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/HeapSemiIndirectPriorityQueue.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import it.unimi.dsi.fastutil.Hash;
import it.unimi.dsi.fastutil.HashCommon;
import static it.unimi.dsi.fastutil.HashCommon.arraySize;
import static it.unimi.dsi.fastutil.HashCommon.maxFill;

import java.util.Map;
import java.util.Arrays;
import java.util.NoSuchElementException;
import java.util.function.Consumer;

import
 VALUE_PACKAGE.VALUE_COLLECTION;
import VALUE_PACKAGE.VALUE_ABSTRACT_COLLECTION;

#if VALUES_PRIMITIVE
import VALUE_PACKAGE.VALUE_ITERATOR;
#endif

#if VALUE_CLASS_Boolean
import it.unimi.dsi.fastutil.booleans.BooleanConsumer;
#endif


#ifdef Linked

import java.util.Comparator;

#if VALUES_PRIMITIVE
import VALUE_PACKAGE.VALUE_LIST_ITERATOR;
#else
import it.unimi.dsi.fastutil.objects.ObjectIterator;
#endif

import it.unimi.dsi.fastutil.objects.AbstractObjectSortedSet;
```

```
import it.unimi.dsi.fastutil.objects.ObjectListIterator;
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;
import it.unimi.dsi.fastutil.objects.ObjectSortedSet;

/**  A type-specific linked hash map with with a fast, small-footprint implementation.
*
* <p>Instances of this class use a hash table to represent a map. The table is
* filled up to a specified <em>load factor</em>, and then doubled in size to
* accommodate new entries. If the table is emptied below <em>one fourth</em>
*
 of the load factor, it is halved in size; however, the table is never reduced to a
* size smaller than that at creation time: this approach makes it
* possible to create maps with a large capacity in which insertions and
* deletions do not cause immediately rehashing. Moreover, halving is
* not performed when deleting entries from an iterator, as it would interfere
* with the iteration process.
*
* <p>Note that {@link #clear()} does not modify the hash table size.
* Rather, a family of {@linkplain #trim() trimming
* methods} lets you control the size of the table; this is particularly useful
* if you reuse instances of this class.
*
* <p>Iterators generated by this map will enumerate pairs in the same order in which they
* have been added to the map (addition of pairs whose key is already present
* in the map does not change the iteration order). Note that this order has nothing in common with the natural
* order of the keys. The order is kept by means of a doubly linked
 list, represented
* <i>via</i> an array of longs parallel to the table.
*
* <p>This class implements the interface of a sorted map, so to allow easy
* access of the iteration order: for instance, you can get the first key
* in iteration order with {@code firstKey()} without having to create an
* iterator; however, this class partially violates the {@link java.util.SortedMap}
* contract because all submap methods throw an exception and {@link
* #comparator()} returns always {@code null}.
*
* <p>Additional methods, such as {@code getAndMoveToFirst()}, make it easy
* to use instances of this class as a cache (e.g., with LRU policy).
*
* <p>The iterators provided by the views of this class using are type-specific
* {@linkplain java.util.ListIterator list iterators}, and can be started at any
* element <em>which is a key of the map</em>, or
* a {@link NoSuchElementException} exception will be thrown.
* If, however, the provided element is not the first or last key in the
 * map, the first access to the list index will require linear time, as in the worst case
* the entire key set must be scanned in iteration order to retrieve the positional
* index of the starting key. If you use just the methods of a type-specific {@link
it.unimi.dsi.fastutil.BidirectionalIterator},
```

```
* however, all operations will be performed in constant time.
*
* @see Hash
* @see HashCommon
*/

public class OPEN_HASH_MAP KEY_VALUE_GENERIC extends ABSTRACT_SORTED_MAP
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable, Hash {

#else

import it.unimi.dsi.fastutil.objects.AbstractObjectSet;
import it.unimi.dsi.fastutil.objects.ObjectIterator;

#ifdef Custom

/** A type-specific hash map with a fast, small-footprint implementation whose {@linkplain
it.unimi.dsi.fastutil.Hash.Strategy hashing strategy}
* is specified at creation time.
*
* <p>Instances of this class use a hash table to represent a map. The table is
* filled up to a specified <em>load factor</em>, and then
 doubled in size to
* accommodate new entries. If the table is emptied below <em>one fourth</em>
* of the load factor, it is halved in size; however, the table is never reduced to a
* size smaller than that at creation time: this approach makes it
* possible to create maps with a large capacity in which insertions and
* deletions do not cause immediately rehashing. Moreover, halving is
* not performed when deleting entries from an iterator, as it would interfere
* with the iteration process.
*
* <p>Note that {@link #clear()} does not modify the hash table size.
* Rather, a family of {@linkplain #trim() trimming
* methods} lets you control the size of the table; this is particularly useful
* if you reuse instances of this class.
*
* @see Hash
* @see HashCommon
*/

public class OPEN_HASH_MAP KEY_VALUE_GENERIC extends ABSTRACT_MAP
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable, Hash {

#else

/** A type-specific hash map with a fast, small-footprint implementation.
 *
* <p>Instances of this class use a hash table to represent a map. The table is
```

```
* filled up to a specified <em>load factor</em>, and then doubled in size to
* accommodate new entries. If the table is emptied below <em>one fourth</em>
* of the load factor, it is halved in size; however, the table is never reduced to a
* size smaller than that at creation time: this approach makes it
* possible to create maps with a large capacity in which insertions and
* deletions do not cause immediately rehashing. Moreover, halving is
* not performed when deleting entries from an iterator, as it would interfere
* with the iteration process.
*
* <p>Note that {@link #clear()} does not modify the hash table size.
* Rather, a family of {@linkplain #trim() trimming
* methods} lets you control the size of the table; this is particularly useful
* if you reuse instances of this class.
*
* @see Hash
* @see HashCommon
*/

public class OPEN_HASH_MAP KEY_VALUE_GENERIC extends ABSTRACT_MAP
 KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable, Hash {

#endif

#endif

 private static final long serialVersionUID = 0L;
 private static final boolean ASSERTS = ASSERTS_VALUE;

 /** The array of keys. */
 protected transient KEY_GENERIC_TYPE[] key;

 /** The array of values. */
 protected transient VALUE_GENERIC_TYPE[] value;

 /** The mask for wrapping a position counter. */
 protected transient int mask;

 /** Whether this map contains the key zero. */
 protected transient boolean containsNullKey;

#ifdef Custom
 /** The hash strategy of this custom map. */
 protected STRATEGY KEY_SUPER_GENERIC strategy;
#endif

#ifdef Linked
 /** The index of the first entry in iteration order. It is valid iff {@link #size} is nonzero; otherwise, it contains -1. */
 protected transient int first = -1;
```

```
 /** The index of the last entry in iteration order. It is valid iff {@link #size} is nonzero; otherwise, it contains -1. */
 protected transient int last = -1;
 /** For each entry,
 the next and the previous entry in iteration order,
  * stored as {@code ((prev & 0xFFFFFFFFL) << 32) | (next & 0xFFFFFFFFL)}.
  * The first entry contains predecessor -1, and the last entry
  * contains successor -1. */
  protected transient long[] link;
#endif

 /** The current table size. */
 protected transient int n;

 /** Threshold after which we rehash. It must be the table size times {@link #f}. */
 protected transient int maxFill;

 /** We never resize below this threshold, which is the construction-time {#n}. */
 protected final transient int minN;

 /** Number of entries in the set (including the key zero, if present). */
 protected int size;

 /** The acceptable load factor. */
 protected final float f;

#ifdef Linked
 /** Cached set of entries. */
 protected transient FastSortedEntrySet KEY_VALUE_GENERIC entries;

 /** Cached set of keys. */
 protected transient SORTED_SET KEY_GENERIC keys;
#else
 /** Cached set of entries. */
 protected transient FastEntrySet KEY_VALUE_GENERIC
 entries;

 /** Cached set of keys. */
 protected transient SET KEY_GENERIC keys;
#endif

 /** Cached collection of values. */
 protected transient VALUE_COLLECTION VALUE_GENERIC values;


#ifdef Custom
 /** Creates a new hash map.
  *
  * <p>The actual table size will be the least power of two greater than {@code expected}/{@code f}.
```

```
  *
  * @param expected the expected number of elements in the hash map.
  * @param f the load factor.
  * @param strategy the strategy.
  */
 SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
 public OPEN_HASH_MAP(final int expected, final float f, final STRATEGY KEY_SUPER_GENERIC strategy) {
  this.strategy = strategy;
#else
 /** Creates a new hash map.
  *
  * <p>The actual table size will be the least power of two greater than {@code expected}/{@code f}.
  *
  * @param expected the expected number of elements in the hash map.
  * @param f the load factor.
  */
 SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
 public OPEN_HASH_MAP(final int expected, final float f) {
#endif
  if
 (f <= 0 || f >= 1) throw new IllegalArgumentException("Load factor must be greater than 0 and smaller than 1");
  if (expected < 0) throw new IllegalArgumentException("The expected number of elements must be nonnegative");

  this.f = f;

  minN = n = arraySize(expected, f);
  mask = n - 1;
  maxFill = maxFill(n, f);
  key = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[n + 1];
  value = VALUE_GENERIC_ARRAY_CAST new VALUE_TYPE[n + 1];
#ifdef Linked
  link = new long[n + 1];
#endif
 }


#ifdef Custom
 /** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor.
  *
  * @param expected the expected number of elements in the hash map.
  * @param strategy the strategy.
  */

 public OPEN_HASH_MAP(final int expected, final STRATEGY KEY_SUPER_GENERIC strategy) {
  this(expected, DEFAULT_LOAD_FACTOR, strategy);
 }
#else
 /** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor.
```

```
  *
  * @param expected the expected number of elements in the hash map.

 */

 public OPEN_HASH_MAP(final int expected) {
  this(expected, DEFAULT_LOAD_FACTOR);
 }
#endif


#ifdef Custom
 /** Creates a new hash map with initial expected {@link Hash#DEFAULT_INITIAL_SIZE} entries
  * and {@link Hash#DEFAULT_LOAD_FACTOR} as load factor.
  * @param strategy the strategy.
  */

 public OPEN_HASH_MAP(final STRATEGY KEY_SUPER_GENERIC strategy) {
  this(DEFAULT_INITIAL_SIZE, DEFAULT_LOAD_FACTOR, strategy);
 }
#else
 /** Creates a new hash map with initial expected {@link Hash#DEFAULT_INITIAL_SIZE} entries
  * and {@link Hash#DEFAULT_LOAD_FACTOR} as load factor.
  */

 public OPEN_HASH_MAP() {
  this(DEFAULT_INITIAL_SIZE, DEFAULT_LOAD_FACTOR);
 }
#endif


#ifdef Custom
 /** Creates a new hash map copying a given one.
  *
  * @param m a {@link Map} to be copied into the new hash map.
  * @param f the load factor.
  * @param strategy the strategy.
  */

 public OPEN_HASH_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> m, final float f, final
 STRATEGY KEY_SUPER_GENERIC strategy) {
  this(m.size(), f, strategy);
  putAll(m);
 }
#else
 /** Creates a new hash map copying a given one.
  *
```

```
  * @param m a {@link Map} to be copied into the new hash map.
  * @param f the load factor.
  */

 public OPEN_HASH_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> m, final float f) {
  this(m.size(), f);
  putAll(m);
 }
#endif

#ifdef Custom
 /** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor copying a given one.
  *
  * @param m a {@link Map} to be copied into the new hash map.
  * @param strategy the strategy.
  */

 public OPEN_HASH_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> m, final STRATEGY KEY_SUPER_GENERIC strategy) {
  this(m, DEFAULT_LOAD_FACTOR, strategy);
 }
#else
 /** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor copying a given one.
  *
  * @param m a {@link Map} to be copied into the new hash map.
  */

 public
 OPEN_HASH_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> m)
{
  this(m, DEFAULT_LOAD_FACTOR);
 }
#endif

#ifdef Custom
 /** Creates a new hash map copying a given type-specific one.
  *
  * @param m a type-specific map to be copied into the new hash map.
  * @param f the load factor.
  * @param strategy the strategy.
  */

 public OPEN_HASH_MAP(final MAP KEY_VALUE_GENERIC m, final float f, final STRATEGY
KEY_SUPER_GENERIC strategy) {
  this(m.size(), f, strategy);
  putAll(m);
 }
```

```
#else
 /** Creates a new hash map copying a given type-specific one.
  *
  * @param m a type-specific map to be copied into the new hash map.
  * @param f the load factor.
  */

 public OPEN_HASH_MAP(final MAP KEY_VALUE_GENERIC m, final float f) {
  this(m.size(), f);
  putAll(m);
 }
#endif

#ifdef Custom
 /** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor copying a given type-
specific one.
  *
  * @param m a type-specific map to be copied into the new
 hash map.
  * @param strategy the strategy.
  */

 public OPEN_HASH_MAP(final MAP KEY_VALUE_GENERIC m, final STRATEGY KEY_SUPER_GENERIC
strategy) {
  this(m, DEFAULT_LOAD_FACTOR, strategy);
 }
#else
 /** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor copying a given type-
specific one.
  *
  * @param m a type-specific map to be copied into the new hash map.
  */

 public OPEN_HASH_MAP(final MAP KEY_VALUE_GENERIC m) {
  this(m, DEFAULT_LOAD_FACTOR);
 }
#endif


#ifdef Custom
 /** Creates a new hash map using the elements of two parallel arrays.
  *
  * @param k the array of keys of the new hash map.
  * @param v the array of corresponding values in the new hash map.
  * @param f the load factor.
  * @param strategy the strategy.
  * @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
  */
```

```
 public OPEN_HASH_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE[] v, final float f,
final STRATEGY KEY_SUPER_GENERIC strategy) {
  this(k.length,
 f, strategy);
  if (k.length != v.length) throw new IllegalArgumentException("The key array and the value array have different
lengths (" + k.length + " and " + v.length + ")");
  for(int i = 0; i < k.length; i++) this.put(k[i], v[i]);
 }
#else
 /** Creates a new hash map using the elements of two parallel arrays.
  *
  * @param k the array of keys of the new hash map.
  * @param v the array of corresponding values in the new hash map.
  * @param f the load factor.
  * @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
  */

 public OPEN_HASH_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE[] v, final float f) {
  this(k.length, f);
  if (k.length != v.length) throw new IllegalArgumentException("The key array and the value array have different
lengths (" + k.length + " and " + v.length + ")");
  for(int i = 0; i < k.length; i++) this.put(k[i], v[i]);
 }
#endif

#ifdef Custom
 /** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR}
 as load factor using the elements of two parallel arrays.
  *
  * @param k the array of keys of the new hash map.
  * @param v the array of corresponding values in the new hash map.
  * @param strategy the strategy.
  * @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
  */

 public OPEN_HASH_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE[] v, final
STRATEGY KEY_SUPER_GENERIC strategy) {
  this(k, v, DEFAULT_LOAD_FACTOR, strategy);
 }
#else
 /** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor using the elements of
two parallel arrays.
  *
  * @param k the array of keys of the new hash map.
  * @param v the array of corresponding values in the new hash map.
  * @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
  */
```

```
 public OPEN_HASH_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE[] v) {
  this(k, v, DEFAULT_LOAD_FACTOR);
 }
#endif



#ifdef Custom
 /** Returns
 the hashing strategy.
  *
  * @return the hashing strategy of this custom hash map.
  */

 public STRATEGY KEY_SUPER_GENERIC strategy() {
  return strategy;
 }
#endif

 private int realSize() {
  return containsNullKey ? size - 1 : size;
 }

 private void ensureCapacity(final int capacity) {
  final int needed = arraySize(capacity, f);
  if (needed > n) rehash(needed);
 }

 private void tryCapacity(final long capacity) {
  final int needed = (int)Math.min(1 << 30, Math.max(2, HashCommon.nextPowerOfTwo((long)Math.ceil(capacity /
f))));
  if (needed > n) rehash(needed);
 }

 private VALUE_GENERIC_TYPE removeEntry(final int pos) {
  final VALUE_GENERIC_TYPE oldValue = value[pos];
#if VALUES_REFERENCE
  value[pos] = null;
#endif
  size--;
#ifdef Linked
  fixPointers(pos);
#endif

  shiftKeys(pos);
  if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
  return oldValue;
```

```
 }

 private VALUE_GENERIC_TYPE removeNullEntry() {
  containsNullKey = false;
#if KEYS_REFERENCE
  key[n]
 = null;
#endif
  final VALUE_GENERIC_TYPE oldValue = value[n];
#if VALUES_REFERENCE
  value[n] = null;
#endif
  size--;
#ifdef Linked
  fixPointers(n);
#endif
  if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
  return oldValue;
 }


 @Override
 public void putAll(Map<? extends KEY_GENERIC_CLASS,? extends VALUE_GENERIC_CLASS> m) {
  if (f <= .5) ensureCapacity(m.size()); // The resulting map will be sized for m.size() elements
  else tryCapacity(size() + m.size()); // The resulting map will be tentatively sized for size() + m.size() elements
  super.putAll(m);
 }

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 private int find(final KEY_GENERIC_TYPE k) {
  if (KEY_EQUALS_NULL(k)) return containsNullKey ? n : -(n + 1);

  KEY_GENERIC_TYPE curr;
  final KEY_GENERIC_TYPE[] key = this.key;
  int pos;

  // The starting point.
  if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) return -(pos + 1);
  if (KEY_EQUALS_NOT_NULL(k, curr)) return pos;
  // There's
 always an unused entry.
  while(true) {
   if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return -(pos + 1);
   if (KEY_EQUALS_NOT_NULL(k, curr)) return pos;
  }
 }
```

```
 private void insert(final int pos, final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
  if (pos == n) containsNullKey = true;
  key[pos] = k;
  value[pos] = v;
#ifdef Linked
  if (size == 0) {
   first = last = pos;
   // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
   link[pos] = -1L;
  }
  else {
   SET_NEXT(link[last], pos);
   SET_UPPER_LOWER(link[pos], last, -1);
   last = pos;
  }
#endif

  if (size++ >= maxFill) rehash(arraySize(size + 1, f));
  if (ASSERTS) checkTable();
 }


 @Override
 public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
  final int pos = find(k);
  if (pos < 0) {
   insert(-pos - 1, k, v);
   return defRetValue;
  }
  final VALUE_GENERIC_TYPE oldValue = value[pos];
  value[pos] = v;
  return oldValue;
 }

#if VALUE_CLASS_Byte || VALUE_CLASS_Short
 || VALUE_CLASS_Char || VALUE_CLASS_Integer || VALUE_CLASS_Long || VALUE_CLASS_Float ||
VALUE_CLASS_Double

 private VALUE_GENERIC_TYPE addToValue(final int pos, final VALUE_GENERIC_TYPE incr) {
  final VALUE_GENERIC_TYPE oldValue = value[pos];
#if VALUE_CLASS_Byte || VALUE_CLASS_Short || VALUE_CLASS_Char
  value[pos] = (VALUE_TYPE)(oldValue + incr);
#else
  value[pos] = oldValue + incr;
#endif
  return oldValue;
 }
```

```
 /** Adds an increment to value currently associated with a key.
  *
  * <p>Note that this method respects the {@linkplain #defaultReturnValue() default return value} semantics: when
  * called with a key that does not currently appears in the map, the key
  * will be associated with the default return value plus
  * the given increment.
  *
  * @param k the key.
  * @param incr the increment.
  * @return the old value, or the {@linkplain #defaultReturnValue() default return value} if no value was present for
the given key.
  */
 public VALUE_GENERIC_TYPE addTo(final
 KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE incr) {
  int pos;

  if (KEY_EQUALS_NULL(k)) {
   if (containsNullKey) return addToValue(n, incr);
   pos = n;
   containsNullKey = true;
  }
  else {
   KEY_GENERIC_TYPE curr;
   final KEY_GENERIC_TYPE[] key = this.key;

   // The starting point.
   if (! KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) {
    if (KEY_EQUALS_NOT_NULL(curr, k)) return addToValue(pos, incr);
    while(! KEY_IS_NULL(curr = key[pos = (pos + 1) & mask]))
     if (KEY_EQUALS_NOT_NULL(curr, k)) return addToValue(pos, incr);
   }
  }

  key[pos] = k;

#if VALUE_CLASS_Byte || VALUE_CLASS_Short || VALUE_CLASS_Char
  value[pos] = (VALUE_TYPE)(defRetValue + incr);
#else
  value[pos] = defRetValue + incr;
#endif

#ifdef Linked
  if (size == 0) {
   first = last = pos;
   // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
   link[pos] = -1L;
  }
  else {
```

```
  SET_NEXT(link[last], pos);
  SET_UPPER_LOWER(link[pos], last, -1);
  last = pos;
  }
#endif

  if (size++
 >= maxFill) rehash(arraySize(size + 1, f));
  if (ASSERTS) checkTable();
  return defRetValue;
  }

#endif

 /** Shifts left entries with the specified hash code, starting at the specified position,
  * and empties the resulting free entry.
  *
  * @param pos a starting position.
  */
 protected final void shiftKeys(int pos) {
  // Shift entries with the same hash.
  int last, slot;
  KEY_GENERIC_TYPE curr;
  final KEY_GENERIC_TYPE[] key = this.key;

  for(;;) {
   pos = ((last = pos) + 1) & mask;

   for(;;) {
    if (KEY_IS_NULL(curr = key[pos])) {
     key[last] = KEY_NULL;
#if VALUES_REFERENCE
     value[last] = null;
#endif
     return;
    }
    slot = KEY2INTHASH(curr) & mask;
    if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
    pos = (pos + 1) & mask;
   }

   key[last] = curr;
   value[last] = value[pos];
#ifdef Linked
   fixPointers(pos, last);
#endif
  }
 }
```

```
@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public VALUE_GENERIC_TYPE REMOVE_VALUE(final
KEY_TYPE k) {
 if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) {
  if (containsNullKey) return removeNullEntry();
  return defRetValue;
 }

 KEY_GENERIC_TYPE curr;
 final KEY_GENERIC_TYPE[] key = this.key;
 int pos;

 // The starting point.
 if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return defRetValue;
 if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return removeEntry(pos);
 while(true) {
  if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return defRetValue;
  if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return removeEntry(pos);
 }
}


#ifdef Linked

 private VALUE_GENERIC_TYPE setValue(final int pos, final VALUE_GENERIC_TYPE v) {
  final VALUE_GENERIC_TYPE oldValue = value[pos];
  value[pos] = v;
  return oldValue;
 }

 /** Removes the mapping associated with the first key in iteration order.
  * @return the value previously associated with the first key in iteration order.
  * @throws NoSuchElementException is this map is empty.
  */
 public VALUE_GENERIC_TYPE
 REMOVE_FIRST_VALUE() {
  if (size == 0) throw new NoSuchElementException();
  final int pos = first;
  // Abbreviated version of fixPointers(pos)
  first = GET_NEXT(link[pos]);
  if (0 <= first) {
   // Special case of SET_PREV(link[first], -1)
   link[first] |= (-1 & 0xFFFFFFFFL) << 32;
  }
  size--;
  final VALUE_GENERIC_TYPE v = value[pos];
```

```
  if (pos == n) {
   containsNullKey = false;
#if KEYS_REFERENCE
   key[n] = null;
#endif
#if VALUES_REFERENCE
   value[n] = null;
#endif
  }
  else shiftKeys(pos);
  if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
  return v;
 }

 /** Removes the mapping associated with the last key in iteration order.
  * @return the value previously associated with the last key in iteration order.
  * @throws NoSuchElementException is this map is empty.
  */
 public VALUE_GENERIC_TYPE REMOVE_LAST_VALUE() {
  if (size == 0) throw new NoSuchElementException();
  final int pos = last;
  // Abbreviated version of fixPointers(pos)
  last
 = GET_PREV(link[pos]);
  if (0 <= last) {
   // Special case of SET_NEXT(link[last], -1)
   link[last] |= -1 & 0xFFFFFFFFL;
  }
  size--;
  final VALUE_GENERIC_TYPE v = value[pos];
  if (pos == n) {
   containsNullKey = false;
#if KEYS_REFERENCE
   key[n] = null;
#endif
#if VALUES_REFERENCE
   value[n] = null;
#endif
  }
  else shiftKeys(pos);
  if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
  return v;
 }

 private void moveIndexToFirst(final int i) {
  if (size == 1 || first == i) return;
  if (last == i) {
   last = GET_PREV(link[i]);
```

```
  // Special case of SET_NEXT(link[last], -1);
  link[last] |= -1 & 0xFFFFFFFFL;
 }
 else {
  final long linki = link[i];
  final int prev = GET_PREV(linki);
  final int next = GET_NEXT(linki);
  COPY_NEXT(link[prev], linki);
  COPY_PREV(link[next], linki);
 }
 SET_PREV(link[first], i);
 SET_UPPER_LOWER(link[i], -1, first);
 first = i;
}

private void moveIndexToLast(final int i) {
 if (size ==
1 ||  last == i) return;
 if (first == i) {
  first = GET_NEXT(link[i]);
  // Special case of SET_PREV(link[first], -1);
  link[first] |= (-1 & 0xFFFFFFFFL) << 32;
 }
 else {
  final long linki = link[i];
  final int prev = GET_PREV(linki);
  final int next = GET_NEXT(linki);
  COPY_NEXT(link[prev], linki);
  COPY_PREV(link[next], linki);
 }
 SET_NEXT(link[last], i);
 SET_UPPER_LOWER(link[i], last, -1);
 last = i;
}

/** Returns the value to which the given key is mapped; if the key is present, it is moved to the first position of the
iteration order.
 *
 * @param k the key.
 * @return the corresponding value, or the {@linkplain #defaultReturnValue() default return value} if no value was
present for the given key.
 */
public VALUE_GENERIC_TYPE getAndMoveToFirst(final KEY_GENERIC_TYPE k) {
 if (KEY_EQUALS_NULL(k)) {
  if (containsNullKey) {
   moveIndexToFirst(n);
   return value[n];
  }
```

```
  return defRetValue;
 }

 KEY_GENERIC_TYPE curr;
 final KEY_GENERIC_TYPE[]
key = this.key;
 int pos;

 // The starting point.
 if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) return defRetValue;
 if (KEY_EQUALS_NOT_NULL(k, curr)) {
  moveIndexToFirst(pos);
  return value[pos];
 }

 // There's always an unused entry.
 while(true) {
  if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return defRetValue;
  if (KEY_EQUALS_NOT_NULL(k, curr))  {
   moveIndexToFirst(pos);
   return value[pos];
  }
 }
}

/** Returns the value to which the given key is mapped; if the key is present, it is moved to the last position of the
iteration order.
 *
 * @param k the key.
 * @return the corresponding value, or the {@linkplain #defaultReturnValue() default return value} if no value was
present for the given key.
 */
public VALUE_GENERIC_TYPE getAndMoveToLast(final KEY_GENERIC_TYPE k) {
 if (KEY_EQUALS_NULL(k)) {
  if (containsNullKey) {
   moveIndexToLast(n);
   return value[n];
  }

  return defRetValue;
 }

 KEY_GENERIC_TYPE
curr;
 final KEY_GENERIC_TYPE[] key = this.key;
 int pos;
```

```
 // The starting point.
 if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) return defRetValue;
 if (KEY_EQUALS_NOT_NULL(k, curr)) {
  moveIndexToLast(pos);
  return value[pos];
 }

 // There's always an unused entry.
 while(true) {
  if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return defRetValue;
  if (KEY_EQUALS_NOT_NULL(k, curr))  {
   moveIndexToLast(pos);
   return value[pos];
  }
 }
}

/** Adds a pair to the map; if the key is already present, it is moved to the first position of the iteration order.
 *
 * @param k the key.
 * @param v the value.
 * @return the old value, or the {@linkplain #defaultReturnValue() default return value} if no value was present for
the given key.
 */
 public VALUE_GENERIC_TYPE putAndMoveToFirst(final KEY_GENERIC_TYPE k, final
VALUE_GENERIC_TYPE v) {
 int pos;

 if (KEY_EQUALS_NULL(k)) {
  if (containsNullKey) {
   moveIndexToFirst(n);
   return setValue(n,
 v);
  }
  containsNullKey = true;
  pos = n;
 }
 else {
  KEY_GENERIC_TYPE curr;
  final KEY_GENERIC_TYPE[] key = this.key;

  // The starting point.
  if (! KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) {
   if (KEY_EQUALS_NOT_NULL(curr, k)) {
    moveIndexToFirst(pos);
    return setValue(pos, v);
   }
   while(! KEY_IS_NULL(curr = key[pos = (pos + 1) & mask]))
```

```
    if (KEY_EQUALS_NOT_NULL(curr, k)) {
     moveIndexToFirst(pos);
     return setValue(pos, v);
    }
   }
  }

  key[pos] = k;
  value[pos] = v;

  if (size == 0) {
   first = last = pos;
   // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
   link[pos] = -1L;
  }
  else {
   SET_PREV(link[first], pos);
   SET_UPPER_LOWER(link[pos], -1, first);
   first = pos;
  }

  if (size++ >= maxFill) rehash(arraySize(size, f));
  if (ASSERTS) checkTable();
  return defRetValue;
 }

 /** Adds a pair to the map; if the key is already present, it is moved to the last position of the iteration order.

 *
  * @param k the key.
  * @param v the value.
  * @return the old value, or the {@linkplain #defaultReturnValue() default return value} if no value was present for
the given key.
  */
 public VALUE_GENERIC_TYPE putAndMoveToLast(final KEY_GENERIC_TYPE k, final
VALUE_GENERIC_TYPE v) {
  int pos;

  if (KEY_EQUALS_NULL(k)) {
   if (containsNullKey) {
    moveIndexToLast(n);
    return setValue(n, v);
   }
   containsNullKey = true;
   pos = n;
  }
  else {
   KEY_GENERIC_TYPE curr;
```

```
  final KEY_GENERIC_TYPE[] key = this.key;

  // The starting point.
  if (! KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) {
   if (KEY_EQUALS_NOT_NULL(curr, k)) {
    moveIndexToLast(pos);
    return setValue(pos, v);
   }
   while(! KEY_IS_NULL(curr = key[pos = (pos + 1) & mask]))
    if (KEY_EQUALS_NOT_NULL(curr, k)) {
     moveIndexToLast(pos);
     return setValue(pos, v);
    }
  }
 }

 key[pos] = k;
 value[pos] = v;

 if (size == 0) {
  first = last = pos;
  // Special case
of SET_UPPER_LOWER(link[pos], -1, -1);
  link[pos] = -1L;
 }
 else {
  SET_NEXT(link[last], pos);
  SET_UPPER_LOWER(link[pos], last, -1);
  last = pos;
 }

 if (size++ >= maxFill) rehash(arraySize(size, f));
 if (ASSERTS) checkTable();
 return defRetValue;
}

#endif

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
 if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) return containsNullKey ? value[n] : defRetValue;

 KEY_GENERIC_TYPE curr;
 final KEY_GENERIC_TYPE[] key = this.key;
 int pos;

 // The starting point.
```

```
 if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return defRetValue;
 if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return value[pos];
 // There's always an unused entry.
 while(true) {
  if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return defRetValue;
  if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return value[pos];
 }
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public boolean containsKey(final
KEY_TYPE k) {
 if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) return containsNullKey;

 KEY_GENERIC_TYPE curr;
 final KEY_GENERIC_TYPE[] key = this.key;
 int pos;

 // The starting point.
 if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return false;
 if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return true;
 // There's always an unused entry.
 while(true) {
  if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return false;
  if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return true;
 }
}


@Override
public boolean containsValue(final VALUE_TYPE v) {
 final VALUE_GENERIC_TYPE value[] = this.value;
 final KEY_GENERIC_TYPE key[] = this.key;
 if (containsNullKey && VALUE_EQUALS(value[n], v)) return true;
 for(int i = n; i-- != 0;) if (! KEY_IS_NULL(key[i]) && VALUE_EQUALS(value[i], v)) return true;
 return false;
}


#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

/** {@inheritDoc} */
@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public VALUE_GENERIC_TYPE getOrDefault(final KEY_TYPE
k, final VALUE_GENERIC_TYPE defaultValue) {
 if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) return containsNullKey ? value[n] : defaultValue;
```

```
 KEY_GENERIC_TYPE curr;
 final KEY_GENERIC_TYPE[] key = this.key;
 int pos;

 // The starting point.
 if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return defaultValue;
 if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return value[pos];
 // There's always an unused entry.
 while(true) {
  if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return defaultValue;
  if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return value[pos];
 }
}

/** {@inheritDoc} */
@Override
public VALUE_GENERIC_TYPE putIfAbsent(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE
v) {
 final int pos = find(k);
 if (pos >= 0) return value[pos];
 insert(-pos - 1, k, v);
 return defRetValue;
}

/** {@inheritDoc} */
@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public boolean remove(final KEY_TYPE k, final VALUE_TYPE v) {
 if (KEY_EQUALS_NULL(KEY_GENERIC_CAST
k)) {
  if (containsNullKey && VALUE_EQUALS(v, value[n])) {
   removeNullEntry();
   return true;
  }
  return false;
 }

 KEY_GENERIC_TYPE curr;
 final KEY_GENERIC_TYPE[] key = this.key;
 int pos;

 // The starting point.
 if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return false;
 if (KEY_EQUALS_NOT_NULL_CAST(k, curr) && VALUE_EQUALS(v, value[pos])) {
  removeEntry(pos);
  return true;
 }
```

```
  while(true) {
   if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return false;
   if (KEY_EQUALS_NOT_NULL_CAST(k, curr) && VALUE_EQUALS(v, value[pos])) {
    removeEntry(pos);
    return true;
   }
  }
 }

 /** {@inheritDoc} */
 @Override
 public boolean replace(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE oldValue, final
VALUE_GENERIC_TYPE v) {
  final int pos = find(k);
  if (pos < 0 || ! VALUE_EQUALS(oldValue, value[pos])) return false;
  value[pos] = v;
  return true;
 }

 /** {@inheritDoc} */
 @Override
 public VALUE_GENERIC_TYPE replace(final
 KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
  final int pos = find(k);
  if (pos < 0) return defRetValue;
  final VALUE_GENERIC_TYPE oldValue = value[pos];
  value[pos] = v;
  return oldValue;
 }

#ifdef JDK_PRIMITIVE_FUNCTION

 /** {@inheritDoc} */
 @Override
 public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_JDK(final KEY_GENERIC_TYPE k, final
JDK_PRIMITIVE_FUNCTION KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) {
  java.util.Objects.requireNonNull(mappingFunction);
  final int pos = find(k);
  if (pos >= 0) return value[pos];
  final VALUE_GENERIC_TYPE newValue =
VALUE_NARROWING(mappingFunction.JDK_PRIMITIVE_FUNCTION_APPLY(k));
  insert(-pos -1, k, newValue);
  return newValue;
 }

#endif

#if KEYS_PRIMITIVE && VALUES_PRIMITIVE
```

```
 /** {@inheritDoc} */
 @Override
 public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_NULLABLE(final KEY_GENERIC_TYPE k, final
JDK_KEY_TO_GENERIC_FUNCTION<? extends VALUE_GENERIC_CLASS> mappingFunction) {
  java.util.Objects.requireNonNull(mappingFunction);
  final
 int pos = find(k);
  if (pos >= 0) return value[pos];
  final VALUE_GENERIC_CLASS newValue = mappingFunction.apply(k);
  if (newValue == null) return defRetValue;
  final VALUE_GENERIC_TYPE v = VALUE_CLASS2TYPE(newValue);
  insert(-pos - 1, k, v);
  return v;
 }

#endif

 /** {@inheritDoc} */
 @Override
 public VALUE_GENERIC_TYPE COMPUTE_IF_PRESENT(final KEY_GENERIC_TYPE k, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) {
  java.util.Objects.requireNonNull(remappingFunction);
  final int pos = find(k);
  if (pos < 0) return defRetValue;
  final VALUE_GENERIC_CLASS newValue = remappingFunction.apply(KEY2OBJ(k),
VALUE2OBJ(value[pos]));
  if (newValue == null) {
   if (KEY_EQUALS_NULL(k)) removeNullEntry();
   else removeEntry(pos);
   return defRetValue;
  }
  return value[pos] = VALUE_CLASS2TYPE(newValue);
 }

 /** {@inheritDoc} */
 @Override
 public VALUE_GENERIC_TYPE COMPUTE(final
 KEY_GENERIC_TYPE k, final java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) {
  java.util.Objects.requireNonNull(remappingFunction);
  final int pos = find(k);
  final VALUE_GENERIC_CLASS newValue = remappingFunction.apply(KEY2OBJ(k), pos >= 0 ?
VALUE2OBJ(value[pos]) : null);
  if (newValue == null) {
   if (pos >= 0) {
    if (KEY_EQUALS_NULL(k)) removeNullEntry();
```

```
    else removeEntry(pos);
   }
   return defRetValue;
  }

  VALUE_GENERIC_TYPE newVal = VALUE_CLASS2TYPE(newValue);
  if (pos < 0) {
   insert(-pos - 1, k, newVal);
   return newVal;
  }

  return value[pos] = newVal;
 }

 /** {@inheritDoc} */
 @Override
 public VALUE_GENERIC_TYPE MERGE(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v,
final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ?
extends VALUE_GENERIC_CLASS> remappingFunction) {
  java.util.Objects.requireNonNull(remappingFunction);

  final
 int pos = find(k);
#if VALUES_PRIMITIVE
  if (pos < 0) {
#else
  if (pos < 0 || value[pos] == null) {
   if (v == null) return defRetValue;
#endif
   insert(-pos - 1, k, v);
   return v;
  }

  final VALUE_GENERIC_CLASS newValue = remappingFunction.apply(VALUE2OBJ(value[pos]),
VALUE2OBJ(v));
  if (newValue == null) {
   if (KEY_EQUALS_NULL(k)) removeNullEntry();
   else removeEntry(pos);
   return defRetValue;
  }

  return value[pos] = VALUE_CLASS2TYPE(newValue);
 }


#endif

 /* Removes all elements from this map.
```

```
 *
 * <p>To increase object reuse, this method does not change the table size.
 * If you want to reduce the table size, you must use {@link #trim()}.
 *
 */
@Override
public void clear() {
 if (size == 0) return;
 size = 0;
 containsNullKey = false;

 Arrays.fill(key, KEY_NULL);
#if VALUES_REFERENCE
 Arrays.fill(value, null);
#endif

#ifdef Linked
 first = last = -1;
#endif
}

@Override
public int size() {
 return size;
}

@Override
public boolean
isEmpty() {
 return size == 0;
}


/** The entry class for a hash map does not record key and value, but
 * rather the position in the hash table of the corresponding entry. This
 * is necessary so that calls to {@link java.util.Map.Entry#setValue(Object)} are reflected in
 * the map */

final class MapEntry implements MAP.Entry KEY_VALUE_GENERIC, Map.Entry<KEY_GENERIC_CLASS,
VALUE_GENERIC_CLASS> {
 // The table index this entry refers to, or -1 if this entry has been deleted.
 int index;

 MapEntry(final int index) {
  this.index = index;
 }

 MapEntry() {}
```

```
  @Override
  public KEY_GENERIC_TYPE ENTRY_GET_KEY() {
      return key[index];
  }

  @Override
  public VALUE_GENERIC_TYPE ENTRY_GET_VALUE() {
   return value[index];
  }

  @Override
  public VALUE_GENERIC_TYPE setValue(final VALUE_GENERIC_TYPE v) {
   final VALUE_GENERIC_TYPE oldValue = value[index];
   value[index] = v;
   return oldValue;
  }

#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use
 the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS getKey() {
   return KEY2OBJ(key[index]);
  }
#endif

#if VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS getValue() {
   return VALUE2OBJ(value[index]);
  }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS setValue(final VALUE_GENERIC_CLASS v) {
   return VALUE2OBJ(setValue(VALUE_CLASS2TYPE(v)));
  }
#endif

  @SuppressWarnings("unchecked")
```

```
  @Override
  public boolean equals(final Object o) {
   if (!(o instanceof Map.Entry)) return false;
   Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS> e = (Map.Entry<KEY_GENERIC_CLASS,
VALUE_GENERIC_CLASS>)o;

   return KEY_EQUALS(key[index], KEY_CLASS2TYPE(e.getKey())) && VALUE_EQUALS(value[index],
VALUE_CLASS2TYPE(e.getValue()));
  }

  @Override
  public
 int hashCode() {
   return KEY2JAVAHASH(key[index]) ^ VALUE2JAVAHASH(value[index]);
  }

  @Override
  public String toString() {
   return key[index] + "=>" + value[index];
  }
 }


#ifdef Linked

 /** Modifies the {@link #link} vector so that the given entry is removed.
  * This method will complete in constant time.
  *
  * @param i the index of an entry.
  */
 protected void fixPointers(final int i) {
  if (size == 0) {
   first = last = -1;
   return;
  }
  if (first == i) {
   first = GET_NEXT(link[i]);
   if (0 <= first) {
    // Special case of SET_PREV(link[first], -1)
    link[first] |= (-1 & 0xFFFFFFFFL) << 32;
   }
   return;
  }
  if (last == i) {
   last = GET_PREV(link[i]);
   if (0 <= last) {
    // Special case of SET_NEXT(link[last], -1)
    link[last] |= -1 & 0xFFFFFFFFL;
```

```
  }
  return;
 }
 final long linki = link[i];
 final int prev = GET_PREV(linki);
 final int next = GET_NEXT(linki);
 COPY_NEXT(link[prev], linki);
 COPY_PREV(link[next],
linki);
}


/** Modifies the {@link #link} vector for a shift from s to d.
 * <p>This method will complete in constant time.
 *
 * @param s the source position.
 * @param d the destination position.
 */
protected void fixPointers(int s, int d) {
 if (size == 1) {
  first = last = d;
  // Special case of SET_UPPER_LOWER(link[d], -1, -1)
  link[d] = -1L;
  return;
 }
 if (first == s) {
  first = d;
  SET_PREV(link[GET_NEXT(link[s])], d);
  link[d] = link[s];
  return;
 }
 if (last == s) {
  last = d;
  SET_NEXT(link[GET_PREV(link[s])], d);
  link[d] = link[s];
  return;
 }
 final long links = link[s];
 final int prev = GET_PREV(links);
 final int next = GET_NEXT(links);
 SET_NEXT(link[prev], d);
 SET_PREV(link[next], d);
 link[d] = links;
}


/** Returns the first key of this map in iteration order.
 *
```

```
 * @return the first key in iteration order.
 */
@Override
public KEY_GENERIC_TYPE FIRST_KEY() {
 if (size == 0) throw new NoSuchElementException();
 return
key[first];
}


/** Returns the last key of this map in iteration order.
 *
 * @return the last key in iteration order.
 */
@Override
public KEY_GENERIC_TYPE LAST_KEY() {
 if (size == 0) throw new NoSuchElementException();
 return key[last];
}

/** {@inheritDoc}
 * <p>This implementation just throws an {@link UnsupportedOperationException}.*/
@Override
public SORTED_MAP KEY_VALUE_GENERIC tailMap(KEY_GENERIC_TYPE from) { throw new
UnsupportedOperationException(); }

/** {@inheritDoc}
 * <p>This implementation just throws an {@link UnsupportedOperationException}.*/
@Override
public SORTED_MAP KEY_VALUE_GENERIC headMap(KEY_GENERIC_TYPE to) { throw new
UnsupportedOperationException(); }

/** {@inheritDoc}
 * <p>This implementation just throws an {@link UnsupportedOperationException}.*/
@Override
public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE
to) { throw new UnsupportedOperationException(); }

/** {@inheritDoc}

* <p>This implementation just returns {@code null}.*/
@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return null; }


/** A list iterator over a linked map.
 *
 * <p>This class provides a list iterator over a linked hash map. The constructor runs in constant time.
```

```
  */
 private class MapIterator {
  /** The entry that will be returned by the next call to {@link java.util.ListIterator#previous()} (or {@code null} if
no previous entry exists). */
  int prev = -1;
  /** The entry that will be returned by the next call to {@link java.util.ListIterator#next()} (or {@code null} if no
next entry exists). */
  int next = -1;
  /** The last entry that was returned (or -1 if we did not iterate or used {@link java.util.Iterator#remove()}). */
  int curr = -1;
  /** The current index (in the sense of a {@link java.util.ListIterator}). Note that this value is not meaningful when
this iterator has been created using the nonempty constructor.*/
  int index = -1;

  protected
 MapIterator() {
   next = first;
   index = 0;
  }

  private MapIterator(final KEY_GENERIC_TYPE from) {
   if (KEY_EQUALS_NULL(from)) {
    if (OPEN_HASH_MAP.this.containsNullKey) {
     next = GET_NEXT(link[n]);
     prev = n;
     return;
    }
    else throw new NoSuchElementException("The key " + from + " does not belong to this map.");
   }

   if (KEY_EQUALS(key[last], from)) {
    prev = last;
    index = size;
    return;
   }

   // The starting point.
   int pos = KEY2INTHASH(from) & mask;

   // There's always an unused entry.
   while(! KEY_IS_NULL(key[pos])) {
    if (KEY_EQUALS_NOT_NULL(key[pos], from)) {
     // Note: no valid index known.
     next = GET_NEXT(link[pos]);
     prev = pos;
     return;
    }
    pos = (pos + 1) & mask;
```

```
 }
 throw new NoSuchElementException("The key " + from + " does not belong to this map.");
}

public boolean hasNext() { return next != -1; }

public boolean hasPrevious() { return prev != -1; }

private final void
ensureIndexKnown() {
 if (index >= 0) return;
 if (prev == -1) {
  index = 0;
  return;
 }
 if (next == -1) {
  index = size;
  return;
 }
 int pos = first;
 index = 1;
 while(pos != prev) {
  pos = GET_NEXT(link[pos]);
  index++;
 }
}

public int nextIndex() {
 ensureIndexKnown();
 return index;
}

public int previousIndex() {
 ensureIndexKnown();
 return index - 1;
}

public int nextEntry() {
 if (! hasNext()) throw new NoSuchElementException();

 curr = next;
 next = GET_NEXT(link[curr]);
 prev = curr;

 if (index >= 0) index++;

 return curr;
}
```

```
 public int previousEntry() {
  if (! hasPrevious()) throw new NoSuchElementException();

  curr = prev;
  prev = GET_PREV(link[curr]);
  next = curr;

  if (index >= 0) index--;

  return curr;
 }

 public void remove() {
  ensureIndexKnown();
  if (curr == -1) throw new IllegalStateException();

  if (curr == prev) {
   /* If the last operation was a next(), we are
removing an entry that preceeds
     the current index, and thus we must decrement it. */
   index--;
   prev = GET_PREV(link[curr]);
  }
  else
   next = GET_NEXT(link[curr]);

  size--;
  /* Now we manually fix the pointers. Because of our knowledge of next
    and prev, this is going to be faster than calling fixPointers(). */
  if (prev == -1) first = next;
  else
   SET_NEXT(link[prev], next);
  if (next == -1) last = prev;
  else
   SET_PREV(link[next], prev);

  int last, slot, pos = curr;
  curr = -1;

  if (pos == n) {
   OPEN_HASH_MAP.this.containsNullKey = false;
#if KEYS_REFERENCE
   key[n] = null;
#endif
#if VALUES_REFERENCE
   value[n] = null;
#endif
```

```
  }
  else {
   KEY_GENERIC_TYPE curr;
   final KEY_GENERIC_TYPE[] key = OPEN_HASH_MAP.this.key;
   // We have to horribly duplicate the shiftKeys() code because we need to update next/prev.
   for(;;) {
    pos = ((last = pos) + 1) & mask;
    for(;;) {
     if (KEY_IS_NULL(curr = key[pos]))
 {
      key[last] = KEY_NULL;
#if VALUES_REFERENCE
      value[last] = null;
#endif
      return;
     }
     slot = KEY2INTHASH(curr) & mask;
     if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
     pos = (pos + 1) & mask;
    }

    key[last] = curr;
    value[last] = value[pos];
    if (next == pos) next = last;
    if (prev == pos) prev = last;
    fixPointers(pos, last);
   }
  }
 }

 public int skip(final int n) {
  int i = n;
  while(i-- != 0 && hasNext()) nextEntry();
  return n - i - 1;
 }

 public int back(final int n) {
  int i = n;
  while(i-- != 0 && hasPrevious()) previousEntry();
  return n - i - 1;
 }

 public void set(@SuppressWarnings("unused") MAP.Entry KEY_VALUE_GENERIC ok) {
  throw new UnsupportedOperationException();
 }

 public void add(@SuppressWarnings("unused") MAP.Entry KEY_VALUE_GENERIC ok) {
  throw new UnsupportedOperationException();
```

```
 }
}

private class EntryIterator extends MapIterator
implements ObjectListIterator<MAP.Entry KEY_VALUE_GENERIC> {
 private MapEntry entry;

 public EntryIterator() {}

 public EntryIterator(KEY_GENERIC_TYPE from) {
  super(from);
 }

 @Override
 public MapEntry next() {
  return entry = new MapEntry(nextEntry());
 }

 @Override
 public MapEntry previous() {
  return entry = new MapEntry(previousEntry());
 }

 @Override
 public void remove() {
  super.remove();
  entry.index = -1; // You cannot use a deleted entry.
 }
}

private class FastEntryIterator extends MapIterator implements ObjectListIterator<MAP.Entry
KEY_VALUE_GENERIC> {
 final MapEntry entry = new MapEntry();

 public FastEntryIterator() {}

 public FastEntryIterator(KEY_GENERIC_TYPE from) {
  super(from);
 }

 @Override
 public MapEntry next() {
  entry.index = nextEntry();
  return entry;
 }

 @Override
 public MapEntry previous() {
```

```
  entry.index = previousEntry();
  return entry;
 }
}

#else

 /** An iterator over a hash map. */

 private
 class MapIterator {
  /** The index of the last entry returned, if positive or zero; initially, {@link #n}. If negative, the last
   entry returned was that of the key of index {@code - pos - 1} from the {@link #wrapped} list. */
  int pos = n;
  /** The index of the last entry that has been returned (more precisely, the value of {@link #pos} if {@link #pos} is
positive,
   or {@link Integer#MIN_VALUE} if {@link #pos} is negative). It is -1 if either
   we did not return an entry yet, or the last returned entry has been removed. */
  int last = -1;
  /** A downward counter measuring how many entries must still be returned. */
  int c = size;
  /** A boolean telling us whether we should return the entry with the null key. */
  boolean mustReturnNullKey = OPEN_HASH_MAP.this.containsNullKey;
  /** A lazily allocated list containing keys of entries that have wrapped around the table because of removals. */
  ARRAY_LIST KEY_GENERIC wrapped;

  public boolean hasNext() {
   return c != 0;
  }

  public
 int nextEntry() {
  if (! hasNext()) throw new NoSuchElementException();

   c--;
   if (mustReturnNullKey) {
    mustReturnNullKey = false;
    return last = n;
   }

   final KEY_GENERIC_TYPE key[] = OPEN_HASH_MAP.this.key;

   for(;;) {
    if (--pos < 0) {
     // We are just enumerating elements from the wrapped list.
     last = Integer.MIN_VALUE;
     final KEY_GENERIC_TYPE k = wrapped.GET_KEY(- pos - 1);
     int p = KEY2INTHASH(k) & mask;
```

```
    while (! KEY_EQUALS_NOT_NULL(k, key[p])) p = (p + 1) & mask;
    return p;
   }
   if (! KEY_IS_NULL(key[pos])) return last = pos;
  }
 }

 /** Shifts left entries with the specified hash code, starting at the specified position,
  * and empties the resulting free entry.
  *
  * @param pos a starting position.
  */
 private void shiftKeys(int pos) {
  // Shift entries with the same hash.
  int last, slot;
  KEY_GENERIC_TYPE curr;
  final KEY_GENERIC_TYPE[] key = OPEN_HASH_MAP.this.key;

  for(;;) {
   pos
= ((last = pos) + 1) & mask;

   for(;;) {
    if (KEY_IS_NULL(curr = key[pos])) {
     key[last] = KEY_NULL;
#if VALUES_REFERENCE
     value[last] = null;
#endif
     return;
    }
    slot = KEY2INTHASH(curr) & mask;
    if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
    pos = (pos + 1) & mask;
   }

   if (pos < last) { // Wrapped entry.
    if (wrapped == null) wrapped = new ARRAY_LIST KEY_GENERIC_DIAMOND(2);
    wrapped.add(key[pos]);
   }

   key[last] = curr;
   value[last] = value[pos];
  }
 }

 public void remove() {
  if (last == -1) throw new IllegalStateException();
  if (last == n) {
```

```
    containsNullKey = false;
#if KEYS_REFERENCE
    key[n] = null;
#endif
#if VALUES_REFERENCE
    value[n] = null;
#endif
   }
   else if (pos >= 0) shiftKeys(last);
   else {
    // We're removing wrapped entries.
#if KEYS_REFERENCE
    OPEN_HASH_MAP.this.REMOVE_VALUE(wrapped.set(- pos - 1, null));
#else
    OPEN_HASH_MAP.this.REMOVE_VALUE(wrapped.GET_KEY(-
 pos - 1));
#endif
    last = -1; // Note that we must not decrement size
    return;
   }

   size--;
   last = -1; // You can no longer remove this entry.
   if (ASSERTS) checkTable();
  }

  public int skip(final int n) {
   int i = n;
   while(i-- != 0 && hasNext()) nextEntry();
   return n - i - 1;
  }
 }


 private class EntryIterator extends MapIterator implements ObjectIterator<MAP.Entry KEY_VALUE_GENERIC>
{
  private MapEntry entry;

  @Override
  public MapEntry next() {
   return entry = new MapEntry(nextEntry());
  }

  @Override
  public void remove() {
   super.remove();
   entry.index = -1; // You cannot use a deleted entry.
  }
```

```
 }

 private class FastEntryIterator extends MapIterator implements ObjectIterator<MAP.Entry
KEY_VALUE_GENERIC> {
  private final MapEntry entry = new MapEntry();

  @Override
  public MapEntry next() {
   entry.index = nextEntry();
   return entry;
  }
 }

#endif

#ifdef Linked
 private final class MapEntrySet extends AbstractObjectSortedSet<MAP.Entry
 KEY_VALUE_GENERIC> implements FastSortedEntrySet KEY_VALUE_GENERIC {

  @Override
  public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() { return new EntryIterator(); }

  @Override
  public Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator() { return null; }

  @Override
  public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> subSet(MAP.Entry KEY_VALUE_GENERIC
fromElement, MAP.Entry KEY_VALUE_GENERIC toElement) { throw new UnsupportedOperationException(); }

  @Override
  public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> headSet(MAP.Entry KEY_VALUE_GENERIC
toElement) { throw new UnsupportedOperationException(); }

  @Override
  public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> tailSet(MAP.Entry KEY_VALUE_GENERIC
fromElement) { throw new UnsupportedOperationException(); }

  @Override
  public MAP.Entry KEY_VALUE_GENERIC first() {
   if (size == 0) throw new NoSuchElementException();
   return new MapEntry(OPEN_HASH_MAP.this.first);
  }

  @Override
  public
 MAP.Entry KEY_VALUE_GENERIC last() {
   if (size == 0) throw new NoSuchElementException();
   return new MapEntry(OPEN_HASH_MAP.this.last);
```

```
  }

#else
 private final class MapEntrySet extends AbstractObjectSet<MAP.Entry KEY_VALUE_GENERIC> implements
FastEntrySet KEY_VALUE_GENERIC {

  @Override
  public ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() { return new EntryIterator(); }

  @Override
  public ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> fastIterator() { return new FastEntryIterator(); }
#endif

  @Override
  SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
  public boolean contains(final Object o) {
   if (!(o instanceof Map.Entry)) return false;
   final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
   if (e.getKey() == null || ! (e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
   if (e.getValue() == null || ! (e.getValue() instanceof VALUE_CLASS)) return false;
#endif
   final KEY_GENERIC_TYPE k = KEY_OBJ2TYPE(KEY_GENERIC_CAST
 e.getKey());
   final VALUE_GENERIC_TYPE v = VALUE_OBJ2TYPE(VALUE_GENERIC_CAST e.getValue());

   if (KEY_EQUALS_NULL(k)) return OPEN_HASH_MAP.this.containsNullKey && VALUE_EQUALS(value[n],
v);

   KEY_GENERIC_TYPE curr;
   final KEY_GENERIC_TYPE[] key = OPEN_HASH_MAP.this.key;
   int pos;

   // The starting point.
   if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) return false;
   if (KEY_EQUALS_NOT_NULL(k, curr)) return VALUE_EQUALS(value[pos], v);
   // There's always an unused entry.
   while(true) {
    if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return false;
    if (KEY_EQUALS_NOT_NULL(k, curr)) return VALUE_EQUALS(value[pos], v);
   }
  }

  @Override
  SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
  public boolean remove(final Object o) {
```

```
   if (!(o instanceof Map.Entry)) return false;
   final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
   if (e.getKey() == null || ! (e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
   if
 (e.getValue() == null || ! (e.getValue() instanceof VALUE_CLASS)) return false;
#endif
   final KEY_GENERIC_TYPE k = KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey());
   final VALUE_GENERIC_TYPE v = VALUE_OBJ2TYPE(VALUE_GENERIC_CAST e.getValue());


   if (KEY_EQUALS_NULL(k)) {
    if (containsNullKey && VALUE_EQUALS(value[n], v)) {
     removeNullEntry();
     return true;
    }
    return false;
   }

   KEY_GENERIC_TYPE curr;
   final KEY_GENERIC_TYPE[] key = OPEN_HASH_MAP.this.key;
   int pos;

   // The starting point.
   if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) return false;
   if (KEY_EQUALS_NOT_NULL(curr, k)) {
    if (VALUE_EQUALS(value[pos], v)) {
     removeEntry(pos);
     return true;
    }
    return false;
   }

   while(true) {
    if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return false;
    if (KEY_EQUALS_NOT_NULL(curr, k))  {
     if (VALUE_EQUALS(value[pos], v)) {
      removeEntry(pos);
      return true;
     }
    }
   }
  }

  @Override
  public
```

```
 int size() {
  return size;
  }

  @Override
  public void clear() {
   OPEN_HASH_MAP.this.clear();
  }

#ifdef Linked
  /** Returns a type-specific list iterator on the elements in this set, starting from a given element of the set.
   * Please see the class documentation for implementation details.
   *
   * @param from an element to start from.
   * @return a type-specific list iterator starting at the given element.
   * @throws IllegalArgumentException if {@code from} does not belong to the set.
   */
  @Override
  public ObjectListIterator<MAP.Entry KEY_VALUE_GENERIC> iterator(final MAP.Entry
KEY_VALUE_GENERIC from) {
   return new EntryIterator(from.ENTRY_GET_KEY());
  }

  /** Returns a type-specific fast list iterator on the elements in this set, starting from the first element.
   * Please see the class documentation for implementation details.
   *
   * @return a type-specific list iterator starting at the first element.
   */
  @Override
  public ObjectListIterator<MAP.Entry
 KEY_VALUE_GENERIC> fastIterator() {
   return new FastEntryIterator();
  }

  /** Returns a type-specific fast list iterator on the elements in this set, starting from a given element of the set.
   * Please see the class documentation for implementation details.
   *
   * @param from an element to start from.
   * @return a type-specific list iterator starting at the given element.
   * @throws IllegalArgumentException if {@code from} does not belong to the set.
   */
  @Override
  public ObjectListIterator<MAP.Entry KEY_VALUE_GENERIC> fastIterator(final MAP.Entry
KEY_VALUE_GENERIC from) {
   return new FastEntryIterator(from.ENTRY_GET_KEY());
  }

  /** {@inheritDoc} */
```

```
  @Override
  public void forEach(final Consumer<? super MAP.Entry KEY_VALUE_GENERIC> consumer) {
   for(int i = size, curr, next = first; i-- != 0;) {
    curr = next;
    next = GET_NEXT(link[curr]);
    consumer.accept(new ABSTRACT_MAP.BasicEntry KEY_VALUE_GENERIC(key[curr], value[curr]));
   }
  }

  /**
 {@inheritDoc} */
  @Override
  public void fastForEach(final Consumer<? super MAP.Entry KEY_VALUE_GENERIC> consumer) {
   final ABSTRACT_MAP.BasicEntry KEY_VALUE_GENERIC entry = new ABSTRACT_MAP.BasicEntry
KEY_VALUE_GENERIC_DIAMOND();
   for(int i = size, curr, next = first; i-- != 0;) {
    curr = next;
    next = GET_NEXT(link[curr]);
    entry.key = key[curr];
    entry.value = value[curr];
    consumer.accept(entry);
   }
  }

#else

  /** {@inheritDoc} */
  @Override
  public void forEach(final Consumer<? super MAP.Entry KEY_VALUE_GENERIC> consumer) {
   if (containsNullKey) consumer.accept(new ABSTRACT_MAP.BasicEntry KEY_VALUE_GENERIC(key[n],
value[n]));
   for(int pos = n; pos-- != 0;)
    if (! KEY_IS_NULL(key[pos])) consumer.accept(new ABSTRACT_MAP.BasicEntry
KEY_VALUE_GENERIC(key[pos], value[pos]));
  }

  /** {@inheritDoc} */
  @Override
  public void fastForEach(final Consumer<? super MAP.Entry KEY_VALUE_GENERIC> consumer) {
   final ABSTRACT_MAP.BasicEntry
 KEY_VALUE_GENERIC entry = new ABSTRACT_MAP.BasicEntry KEY_VALUE_GENERIC_DIAMOND();
   if (containsNullKey) {
    entry.key = key[n];
    entry.value = value[n];
    consumer.accept(entry);
   }
   for(int pos = n; pos-- != 0;)
    if (! KEY_IS_NULL(key[pos])) {
```

```
    entry.key = key[pos];
    entry.value = value[pos];
    consumer.accept(entry);
   }
  }

#endif
 }


#ifdef Linked
 @Override
 public FastSortedEntrySet KEY_VALUE_GENERIC ENTRYSET() {
  if (entries == null) entries = new MapEntrySet();
#else
 @Override
 public FastEntrySet KEY_VALUE_GENERIC ENTRYSET() {
  if (entries == null) entries = new MapEntrySet();
#endif
  return entries;
 }

 /** An iterator on keys.
  *
  * <p>We simply override the {@link java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()}
methods
  * (and possibly their type-specific counterparts) so that they return keys
  * instead of entries.
  */

#ifdef Linked
 private final class KeyIterator extends MapIterator implements KEY_LIST_ITERATOR
 KEY_GENERIC {
  public KeyIterator(final KEY_GENERIC_TYPE k) { super(k); }

  @Override
  public KEY_GENERIC_TYPE PREV_KEY() { return key[previousEntry()]; }

#else
 private final class KeyIterator extends MapIterator implements KEY_ITERATOR KEY_GENERIC {
#endif
  public KeyIterator() { super(); }

  @Override
  public KEY_GENERIC_TYPE NEXT_KEY() { return key[nextEntry()]; }
 }
```

```
#ifdef Linked
 private final class KeySet extends ABSTRACT_SORTED_SET KEY_GENERIC {

  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
KeyIterator(from); }

  @Override
  public KEY_LIST_ITERATOR KEY_GENERIC iterator() { return new KeyIterator(); }
#else
 private final class KeySet extends ABSTRACT_SET KEY_GENERIC {

  @Override
  public KEY_ITERATOR KEY_GENERIC iterator() { return new KeyIterator(); }
#endif

  /** {@inheritDoc} */
  @Override
#ifdef JDK_PRIMITIVE_KEY_CONSUMER
  public void forEach(final JDK_PRIMITIVE_KEY_CONSUMER consumer) {
#else
  public
 void forEach(final KEY_CONSUMER KEY_SUPER_GENERIC consumer) {
#endif
   if (containsNullKey) consumer.accept(key[n]);
   for(int pos = n; pos-- != 0;) {
    final KEY_GENERIC_TYPE k = key[pos];
    if (! KEY_IS_NULL(k)) consumer.accept(k);
   }
  }

  @Override
  public int size() { return size; }

  @Override
  public boolean contains(KEY_TYPE k) { return containsKey(k); }

  @Override
  public boolean remove(KEY_TYPE k) {
   final int oldSize = size;
   OPEN_HASH_MAP.this.REMOVE_VALUE(k);
   return size != oldSize;
  }

  @Override
  public void clear() { OPEN_HASH_MAP.this.clear();}
```

```
#ifdef Linked
  @Override
  public KEY_GENERIC_TYPE FIRST() {
   if (size == 0) throw new NoSuchElementException();
   return key[first];
  }

  @Override
  public KEY_GENERIC_TYPE LAST() {
   if (size == 0) throw new NoSuchElementException();
   return key[last];
  }

  @Override
  public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return null; }

  @Override
  public SORTED_SET KEY_GENERIC
 tailSet(KEY_GENERIC_TYPE from) { throw new UnsupportedOperationException(); }

  @Override
  public SORTED_SET KEY_GENERIC headSet(KEY_GENERIC_TYPE to) { throw new
UnsupportedOperationException(); }

  @Override
  public SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE to) { throw
new UnsupportedOperationException(); }
#endif
 }


#ifdef Linked
 @Override
 public SORTED_SET KEY_GENERIC keySet() {
#else
 @Override
 public SET KEY_GENERIC keySet() {
#endif
  if (keys == null) keys = new KeySet();
  return keys;
 }


 /** An iterator on values.
  *
  * <p>We simply override the {@link java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()}
methods
  * (and possibly their type-specific counterparts) so that they return values
```

```
  * instead of entries.
  */

#ifdef Linked
 private final class ValueIterator extends MapIterator implements VALUE_LIST_ITERATOR VALUE_GENERIC {
  @Override
  public VALUE_GENERIC_TYPE PREV_VALUE() { return value[previousEntry()];
 }

#else
 private final class ValueIterator extends MapIterator implements VALUE_ITERATOR VALUE_GENERIC {
#endif
  public ValueIterator() { super(); }

  @Override
  public VALUE_GENERIC_TYPE NEXT_VALUE() { return value[nextEntry()]; }
 }

 @Override
 public VALUE_COLLECTION VALUE_GENERIC values() {
  if (values == null) values = new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {
    @Override
    public VALUE_ITERATOR VALUE_GENERIC iterator() { return new ValueIterator(); }
    @Override
    public int size() { return size; }
    @Override
    public boolean contains(VALUE_TYPE v) { return containsValue(v); }
    @Override
    public void clear() { OPEN_HASH_MAP.this.clear(); }

    /** {@inheritDoc} */
    @Override
#ifdef JDK_PRIMITIVE_VALUE_CONSUMER
    public void forEach(final JDK_PRIMITIVE_VALUE_CONSUMER consumer) {
#else
    public void forEach(final VALUE_CONSUMER VALUE_SUPER_GENERIC consumer) {
#endif
     if (containsNullKey) consumer.accept(value[n]);
     for(int pos =
 n; pos-- != 0;)
      if (! KEY_IS_NULL(key[pos])) consumer.accept(value[pos]);
    }

   };

  return values;
 }
```

```
/** Rehashes the map, making the table as small as possible.
 *
 * <p>This method rehashes the table to the smallest size satisfying the
 * load factor. It can be used when the set will not be changed anymore, so
 * to optimize access speed and size.
 *
 * <p>If the table size is already the minimum possible, this method
 * does nothing.
 *
 * @return true if there was enough memory to trim the map.
 * @see #trim(int)
 */

public boolean trim() {
 return trim(size);
}


/** Rehashes this map if the table is too large.
 *
 * <p>Let <var>N</var> be the smallest table size that can hold
 * <code>max(n,{@link #size()})</code> entries, still satisfying the load factor. If the current
 * table size is smaller than or equal to <var>N</var>, this method does
 * nothing. Otherwise, it rehashes this map in a table of size
 * <var>N</var>.
 *
 * <p>This method
is useful when reusing maps.  {@linkplain #clear() Clearing a
 * map} leaves the table size untouched. If you are reusing a map
 * many times, you can call this method with a typical
 * size to avoid keeping around a very large table just
 * because of a few large transient maps.
 *
 * @param n the threshold for the trimming.
 * @return true if there was enough memory to trim the map.
 * @see #trim()
 */

public boolean trim(final int n) {
 final int l = HashCommon.nextPowerOfTwo((int)Math.ceil(n / f));
 if (l >= this.n || size > maxFill(l, f)) return true;
 try {
  rehash(l);
 }
 catch(OutOfMemoryError cantDoIt) { return false; }
 return true;
}
```

```
/** Rehashes the map.
 *
 * <p>This method implements the basic rehashing strategy, and may be
 * overridden by subclasses implementing different rehashing strategies (e.g.,
 * disk-based rehashing). However, you should not override this method
 * unless you understand the internal workings of this class.
 *
 * @param
newN the new size
 */

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
protected void rehash(final int newN) {
 final KEY_GENERIC_TYPE key[] = this.key;
 final VALUE_GENERIC_TYPE value[] = this.value;

 final int mask = newN - 1; // Note that this is used by the hashing macro
 final KEY_GENERIC_TYPE newKey[] = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[newN + 1];
 final VALUE_GENERIC_TYPE newValue[] = VALUE_GENERIC_ARRAY_CAST new VALUE_TYPE[newN +
1];

#ifdef Linked
 int i = first, prev = -1, newPrev = -1, t, pos;
 final long link[] = this.link;
 final long newLink[] = new long[newN + 1];
 first = -1;

 for(int j = size; j-- != 0;) {
  if (KEY_EQUALS_NULL(key[i])) pos = newN;
  else {
   pos = KEY2INTHASH(key[i]) & mask;
   while (! KEY_IS_NULL(newKey[pos])) pos = (pos + 1) & mask;
  }

  newKey[pos] = key[i];
  newValue[pos] = value[i];

  if (prev != -1) {
   SET_NEXT(newLink[newPrev], pos);
   SET_PREV(newLink[pos], newPrev);
   newPrev = pos;
  }
  else {
   newPrev
= first = pos;
   // Special case of SET(newLink[pos], -1, -1);
   newLink[pos] = -1L;
```

```
    }

    t = i;
    i = GET_NEXT(link[i]);
    prev = t;
   }

   this.link = newLink;
   this.last = newPrev;
   if (newPrev != -1)
    // Special case of SET_NEXT(newLink[newPrev], -1);
    newLink[newPrev] |= -1 & 0xFFFFFFFFL;
#else
   int i = n, pos;

   for(int j = realSize(); j-- != 0;) {
    while(KEY_IS_NULL(key[--i]));

    if (! KEY_IS_NULL(newKey[pos = KEY2INTHASH(key[i]) & mask]))
     while (! KEY_IS_NULL(newKey[pos = (pos + 1) & mask]));

    newKey[pos] = key[i];
    newValue[pos] = value[i];
   }

   newValue[newN] = value[n];

#endif

   n = newN;
   this.mask = mask;
   maxFill = maxFill(n, f);
   this.key = newKey;
   this.value = newValue;
  }


  /** Returns a deep copy of this map.
   *
   * <p>This method performs a deep copy of this hash map; the data stored in the
   * map, however, is not cloned. Note that this makes a difference only for object keys.
   *
   *  @return a deep copy of this map.
   */
  @Override
  SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
  public
```

```
 OPEN_HASH_MAP KEY_VALUE_GENERIC clone() {
  OPEN_HASH_MAP KEY_VALUE_GENERIC c;
  try {
   c = (OPEN_HASH_MAP KEY_VALUE_GENERIC)super.clone();
  }
  catch(CloneNotSupportedException cantHappen) {
   throw new InternalError();
  }

  c.keys = null;
  c.values = null;
  c.entries = null;
  c.containsNullKey = containsNullKey;

  c.key = key.clone();
  c.value = value.clone();
#ifdef Linked
  c.link = link.clone();
#endif
#ifdef Custom
  c.strategy = strategy;
#endif
  return c;
 }


 /** Returns a hash code for this map.
  *
  * This method overrides the generic method provided by the superclass.
  * Since {@code equals()} is not overriden, it is important
  * that the value returned by this method is the same value as
  * the one returned by the overriden method.
  *
  * @return a hash code for this map.
  */

 @Override
 public int hashCode() {
  int h = 0;
  for(int j = realSize(), i = 0, t = 0; j-- != 0;) {
   while(KEY_IS_NULL(key[i]))
 i++;
#if KEYS_REFERENCE
   if (this != key[i])
#endif
    t = KEY2JAVAHASH_NOT_NULL(key[i]);
#if VALUES_REFERENCE
   if (this != value[i])
```

```
#endif
   t ^=  VALUE2JAVAHASH(value[i]);
  h += t;
  i++;
 }
 // Zero / null keys have hash zero.
 if (containsNullKey) h += VALUE2JAVAHASH(value[n]);
 return h;
}



private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
 final KEY_GENERIC_TYPE key[] = this.key;
 final VALUE_GENERIC_TYPE value[] = this.value;
 final MapIterator i = new MapIterator();

 s.defaultWriteObject();

 for(int j = size, e; j-- != 0;) {
  e = i.nextEntry();
  s.WRITE_KEY(key[e]);
  s.WRITE_VALUE(value[e]);
 }
}



SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
 s.defaultReadObject();

 n = arraySize(size, f);
 maxFill = maxFill(n, f);
 mask = n - 1;

 final KEY_GENERIC_TYPE key[] = this.key = KEY_GENERIC_ARRAY_CAST
new KEY_TYPE[n + 1];
 final VALUE_GENERIC_TYPE value[] = this.value = VALUE_GENERIC_ARRAY_CAST new
VALUE_TYPE[n + 1];
#ifdef Linked
 final long link[] = this.link = new long[n + 1];
 int prev = -1;
 first = last = -1;
#endif

 KEY_GENERIC_TYPE k;
 VALUE_GENERIC_TYPE v;
```

```
  for(int i = size, pos; i-- != 0;) {

   k = KEY_GENERIC_CAST s.READ_KEY();
   v = VALUE_GENERIC_CAST s.READ_VALUE();

   if (KEY_EQUALS_NULL(k)) {
    pos = n;
    containsNullKey = true;
   }
   else {
    pos = KEY2INTHASH(k) & mask;
    while (! KEY_IS_NULL(key[pos])) pos = (pos + 1) & mask;
   }

   key[pos] = k;
   value[pos] = v;

#ifdef Linked
   if (first != -1) {
    SET_NEXT(link[prev], pos);
    SET_PREV(link[pos], prev);
    prev = pos;
   }
   else {
    prev = first = pos;
    // Special case of SET_PREV(newLink[pos], -1);
    link[pos] |= (-1L & 0xFFFFFFFFL) << 32;
   }
#endif
  }

#ifdef Linked
  last = prev;
  if (prev != -1)
   // Special case of SET_NEXT(link[prev], -1);
   link[prev] |= -1 &
 0xFFFFFFFFL;
#endif

  if (ASSERTS) checkTable();
  }



#ifdef ASSERTS_CODE
 private void checkTable() {
  assert (n & -n) == n : "Table length is not a power of two: " + n;
```

```
  assert n == key.length - 1;
  int n = key.length - 1;
  while(n-- != 0)
   if (! KEY_IS_NULL(key[n]) && ! containsKey(key[n]))
    throw new AssertionError("Hash table has key " + key[n] + " marked as occupied, but the key does not belong to the
table");

#if KEYS_PRIMITIVE
  java.util.HashSet<KEY_GENERIC_CLASS> s = new java.util.HashSet<KEY_GENERIC_CLASS> ();
#else
  java.util.HashSet<Object> s = new java.util.HashSet<Object>();
#endif

  for(int i = key.length; i-- != 0;)
   if (! KEY_IS_NULL(key[i]) && ! s.add(key[i])) throw new AssertionError("Key " + key[i] + " appears twice at
position " + i);

#ifdef Linked
  KEY_BIDI_ITERATOR KEY_GENERIC i = keySet().iterator();
  KEY_GENERIC_TYPE k;
  n = size();
  while(n-- != 0)
   if (! containsKey(k = i.NEXT_KEY()))
    throw new AssertionError("Linked hash table forward
 enumerates key " + k + ", but the key does not belong to the table");

  if (i.hasNext()) throw new AssertionError("Forward iterator not exhausted");

  n = size();
  if (n > 0) {
   i = keySet().iterator(LAST_KEY());
   while(n-- != 0)
    if (! containsKey(k = i.PREV_KEY()))
     throw new AssertionError("Linked hash table backward enumerates key " + k + ", but the key does not belong to the
table");

   if (i.hasPrevious()) throw new AssertionError("Previous iterator not exhausted");
  }
#endif
 }
#else
 private void checkTable() {}
#endif


#ifdef TEST

 private static long seed = System.currentTimeMillis();
```

```
 private static java.util.Random r = new java.util.Random(seed);

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif !KEY_CLASS_Reference
#ifdef Custom
  int i = r.nextInt(3);
  byte a[] = new byte[i];
  while(i-- != 0) a[i] = (byte)r.nextInt();
  return
 a;
#else
  return Integer.toBinaryString(r.nextInt());
#endif
#else
  return new java.io.Serializable() {};
#endif
 }

 private static VALUE_TYPE genValue() {
#if VALUE_CLASS_Byte || VALUE_CLASS_Short || VALUE_CLASS_Character
  return (VALUE_TYPE)(r.nextInt());
#elif VALUES_PRIMITIVE
  return r.NEXT_VALUE();
#elif !VALUE_CLASS_Reference
  return Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }

 private static final class ArrayComparator implements java.util.Comparator {
  public int compare(Object a, Object b) {
   byte[] aa = (byte[])a;
   byte[] bb = (byte[])b;
   int length = Math.min(aa.length, bb.length);
   for(int i = 0; i < length; i++) {
    if (aa[i] < bb[i]) return -1;
    if (aa[i] > bb[i]) return 1;
   }
   return aa.length == bb.length ? 0 : (aa.length < bb.length ? -1 : 1);
  }
 }

 private static final class MockMap extends java.util.TreeMap {
```

```
 private java.util.List list = new java.util.ArrayList();

 public MockMap(java.util.Comparator
c) { super(c); }

 public Object put(Object k, Object v) {
  if (! containsKey(k)) list.add(k);
  return super.put(k, v);
 }

 public void putAll(Map m) {
  java.util.Iterator i = m.entrySet().iterator();
  while(i.hasNext()) {
   Map.Entry e = (Map.Entry)i.next();
   put(e.getKey(), e.getValue());
  }
 }

 public Object remove(Object k) {
  if (containsKey(k)) {
   int i = list.size();
   while(i-- != 0) if (comparator().compare(list.get(i), k) == 0) {
    list.remove(i);
    break;
   }
  }
  return super.remove(k);
 }

 private void justRemove(Object k) { super.remove(k); }
 private java.util.Set justEntrySet() { return super.entrySet(); }
 private java.util.Set justKeySet() { return super.keySet(); }

 public java.util.Set keySet() {
  return new java.util.AbstractSet() {
    final java.util.Set keySet = justKeySet();

    public boolean contains(Object k) { return keySet.contains(k); }
    public int size() { return keySet.size();
}
    public java.util.Iterator iterator() {
     return new java.util.Iterator() {
       final java.util.Iterator iterator = list.iterator();
       Object curr;
       public Object next() { return curr = iterator.next(); }
       public boolean hasNext() { return iterator.hasNext(); }
       public void remove() {
        justRemove(curr);
```

```
        iterator.remove();
       }
      };

    }
   };

  }

  public java.util.Set entrySet() {
   return new java.util.AbstractSet() {
    final java.util.Set entrySet = justEntrySet();

    public boolean contains(Object k) { return entrySet.contains(k); }
    public int size() { return entrySet.size(); }
    public java.util.Iterator iterator() {
     return new java.util.Iterator() {
       final java.util.Iterator iterator = list.iterator();
       Object curr;
       public Object next() {
        curr = iterator.next();
#if VALUE_CLASS_Reference
#if KEY_CLASS_Reference
        return new ABSTRACT_MAP.BasicEntry((Object)curr,
 (Object)get(curr)) {
#else
        return new ABSTRACT_MAP.BasicEntry((KEY_CLASS)curr, (Object)get(curr)) {
#endif
#else
#if KEY_CLASS_Reference
        return new ABSTRACT_MAP.BasicEntry((Object)curr, (VALUE_CLASS)get(curr)) {
#else
        return new ABSTRACT_MAP.BasicEntry((KEY_CLASS)curr, (VALUE_CLASS)get(curr)) {
#endif
#endif
          public VALUE_TYPE setValue(VALUE_TYPE v) {
           return VALUE_OBJ2TYPE(put(getKey(), VALUE2OBJ(v)));
          }
         };
       }
       public boolean hasNext() { return iterator.hasNext(); }
       public void remove() {
        justRemove(((Map.Entry)curr).getKey());
        iterator.remove();
       }
      };
    }
   };
```

```
 }
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

private static String format(double d) {
 StringBuffer s = new StringBuffer();
 return format.format(d, s,
fp).toString();
}


private static void speedTest(int n, float f, boolean comp) {
#ifndef Custom
 int i, j;
 OPEN_HASH_MAP m;
#ifdef Linked
 java.util.LinkedHashMap t;
#else
 java.util.HashMap t;
#endif
 KEY_TYPE k[] = new KEY_TYPE[n];
 KEY_TYPE nk[] = new KEY_TYPE[n];
 VALUE_TYPE v[] = new VALUE_TYPE[n];
 long ns;

 for(i = 0; i < n; i++) {
  k[i] = genKey();
  nk[i] = genKey();
  v[i] = genValue();
 }

 double totPut = 0, totYes = 0, totNo = 0, totIter = 0, totRemYes = 0, totRemNo = 0, d;

 if (comp) { for(j = 0; j < 20; j++) {

#ifdef Linked
  t = new java.util.LinkedHashMap(16);
#else
  t = new java.util.HashMap(16);
#endif

  /* We put pairs to t. */
  ns = System.nanoTime();
  for(i = 0; i < n;  i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));
  d = (System.nanoTime() - ns) / (double)n;
  if (j > 2) totPut += d;
```

```
  System.out.print("Put: " + format(d) + "ns ");

  /* We check for pairs in t. */
  ns = System.nanoTime();
  for(i = 0; i < n;  i++) t.containsKey(KEY2OBJ(k[i]));
  d
= (System.nanoTime() - ns) / (double)n;
  if (j > 2) totYes += d;
  System.out.print("Yes: " + format(d) + "ns ");

  /* We check for pairs not in t. */
  ns = System.nanoTime();
  for(i = 0; i < n;  i++) t.containsKey(KEY2OBJ(nk[i]));
  d = (System.nanoTime() - ns) / (double)n;
  if (j > 2) totNo += d;
  System.out.print("No: " + format(d) + "ns ");

  /* We iterate on t. */
  ns = System.nanoTime();
  for(java.util.Iterator it = t.entrySet().iterator(); it.hasNext(); it.next());
  d = (System.nanoTime() - ns) / (double)n;
  if (j > 2) totIter += d;
  System.out.print("Iter: " + format(d) + "ns ");

  /* We delete pairs not in t. */
  ns = System.nanoTime();
  for(i = 0; i < n;  i++) t.remove(KEY2OBJ(nk[i]));
  d = (System.nanoTime() - ns) / (double)n;
  if (j > 2) totRemNo += d;
  System.out.print("RemNo: " + format(d) + "ns ");

  /* We delete pairs in t. */
  ns = System.nanoTime();
  for(i = 0; i < n;  i++) t.remove(KEY2OBJ(k[i]));
  d
= (System.nanoTime() - ns) / (double)n;
  if (j > 2) totRemYes += d;
  System.out.print("RemYes: " + format(d) + "ns ");

  System.out.println();
  }

  System.out.println();
  System.out.println("java.util Put: " + format(totPut/(j-3)) + "ns Yes: " + format(totYes/(j-3)) + "ns No: " +
format(totNo/(j-3)) + "ns Iter: " + format(totIter/(j-3)) + "ns RemNo: " + format(totRemNo/(j-3)) + "ns RemYes: " +
format(totRemYes/(j-3)) + "K/s");

  System.out.println();
```

```
totPut = totYes = totNo = totIter = totRemYes = totRemNo = 0;

}

for(j = 0; j < 20; j++) {

 m = new OPEN_HASH_MAP(16, f);

 /* We put pairs to m. */
 ns = System.nanoTime();
 for(i = 0; i < n;  i++) m.put(k[i], v[i]);
 d = (System.nanoTime() - ns) / (double)n;
 if (j > 2) totPut += d;
 System.out.print("Put: " + format(d) + "ns ");

 /* We check for pairs in m. */
 ns = System.nanoTime();
 for(i = 0; i < n;  i++) m.containsKey(k[i]);
 d = (System.nanoTime() - ns) / (double)n;
 if (j > 2) totYes
+= d;
 System.out.print("Yes: " + format(d) + "ns ");

 /* We check for pairs not in m. */
 ns = System.nanoTime();
 for(i = 0; i < n;  i++) m.containsKey(nk[i]);
 d = (System.nanoTime() - ns) / (double)n;
 if (j > 2) totNo += d;
 System.out.print("No: " + format(d) + "ns ");

 /* We iterate on m. */
 ns = System.nanoTime();
 for(java.util.Iterator it = m.entrySet().iterator(); it.hasNext(); it.next());
 d = (System.nanoTime() - ns) / (double)n;
 if (j > 2) totIter += d;
 System.out.print("Iter: " + format(d) + "ns ");

 /* We delete pairs not in m. */
 ns = System.nanoTime();
 for(i = 0; i < n;  i++) m.remove(nk[i]);
 d = (System.nanoTime() - ns) / (double)n;
 if (j > 2) totRemNo += d;
 System.out.print("RemNo: " + format(d) + "ns ");

 /* We delete pairs in m. */
 ns = System.nanoTime();
 for(i = 0; i < n;  i++) m.remove(k[i]);
```

```
  d = (System.nanoTime() - ns) / (double)n;
  if (j > 2) totRemYes += d;
  System.out.print("RemYes: " + format(d)
 + "ns ");

  System.out.println();
  }


  System.out.println();
  System.out.println("fastutil  Put: " + format(totPut/(j-3)) + "ns Yes: " + format(totYes/(j-3)) + "ns No: " +
format(totNo/(j-3)) + "ns Iter: " + format(totIter/(j-3)) + "ns RemNo: " + format(totRemNo/(j-3)) + "ns RemYes: " +
format(totRemYes/(j-3)) + "ns");

  System.out.println();
#endif
 }

 private static boolean valEquals(Object o1, Object o2) {
  return o1 == null ? o2 == null : o1.equals(o2);
 }

 private static void fatal(String msg) {
  System.out.println(msg);
  System.exit(1);
 }

 private static void ensure(boolean cond, String msg) {
  if (cond) return;
  fatal(msg);
 }

 protected static void runTest(int n, float f) {
#if !defined(Custom) || KEYS_REFERENCE

#ifdef Custom
  OPEN_HASH_MAP m = new OPEN_HASH_MAP(Hash.DEFAULT_INITIAL_SIZE, f,
it.unimi.dsi.fastutil.bytes.ByteArrays.HASH_STRATEGY);
#else
  OPEN_HASH_MAP m = new OPEN_HASH_MAP(Hash.DEFAULT_INITIAL_SIZE, f);
#endif

#ifdef Linked
#ifdef Custom
  Map
 t = new MockMap(new ArrayComparator());
#else
  Map t = new java.util.LinkedHashMap();
```

```
#endif
#else
#ifdef Custom
  Map t = new java.util.TreeMap(new ArrayComparator());
#else
  Map t = new java.util.HashMap();
#endif
#endif

  /* First of all, we fill t with random data. */

  for(int i=0; i<n;  i++) t.put(KEY2OBJ(genKey()), VALUE2OBJ(genValue()));

  /* Now we add to m the same data */

  m.putAll(t);

  if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after insertion");
  if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after insertion");

  /* Now we check that m actually holds that data. */

  for(java.util.Iterator i=t.entrySet().iterator(); i.hasNext();) {
   java.util.Map.Entry e = (java.util.Map.Entry)i.next();
   if (!valEquals(e.getValue(), m.get(e.getKey())))
    System.out.println("Error (" + seed + "): m and t differ on an entry ("+e+") after insertion (iterating on t)");
  }

  /* Now we check that m actually holds that
 data, but iterating on m. */

  for(java.util.Iterator i=m.entrySet().iterator(); i.hasNext();) {
   java.util.Map.Entry e = (java.util.Map.Entry)i.next();
   if (!valEquals(e.getValue(), t.get(e.getKey())))
    System.out.println("Error (" + seed + "): m and t differ on an entry ("+e+") after insertion (iterating on m)");
  }

  /* Now we check that m actually holds the same keys. */

  for(java.util.Iterator i=t.keySet().iterator(); i.hasNext();) {
   Object o = i.next();
   if (!m.containsKey(o)) {
    System.out.println("Error (" + seed + "): m and t differ on a key ("+o+") after insertion (iterating on t)");
    System.exit(1);
   }
   if (!m.keySet().contains(o)) {
    System.out.println("Error (" + seed + "): m and t differ on a key ("+o+", in keySet()) after insertion (iterating on t)");
    System.exit(1);
```

```
 }
}

/* Now we check that m actually holds the same keys, but iterating on m. */

for(java.util.Iterator i=m.keySet().iterator(); i.hasNext();) {
 Object o = i.next();
 if
(!t.containsKey(o)) {
  System.out.println("Error (" + seed + "): m and t differ on a key after insertion (iterating on m)");
  System.exit(1);
 }
 if (!t.keySet().contains(o)) {
  System.out.println("Error (" + seed + "): m and t differ on a key (in keySet()) after insertion (iterating on m)");
  System.exit(1);
 }
}


/* Now we check that m actually hold the same values. */

for(java.util.Iterator i=t.values().iterator(); i.hasNext();) {
 Object o = i.next();
 if (!m.containsValue(o)) {
  System.out.println("Error (" + seed + "): m and t differ on a value after insertion (iterating on t)");
  System.exit(1);
 }
 if (!m.values().contains(o)) {
  System.out.println("Error (" + seed + "): m and t differ on a value (in values()) after insertion (iterating on t)");
  System.exit(1);
 }
}

/* Now we check that m actually hold the same values, but iterating on m. */

for(java.util.Iterator i=m.values().iterator(); i.hasNext();) {
 Object
o = i.next();
 if (!t.containsValue(o)) {
  System.out.println("Error (" + seed + "): m and t differ on a value after insertion (iterating on m)");
  System.exit(1);
 }
 if (!t.values().contains(o)) {
  System.out.println("Error (" + seed + "): m and t differ on a value (in values()) after insertion (iterating on m)");
  System.exit(1);
 }
}
```

```
  /* Now we check that inquiries about random data give the same answer in m and t. For
     m we use the polymorphic method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();
   if (m.containsKey(KEY2OBJ(T)) != t.containsKey(KEY2OBJ(T))) {
    System.out.println("Error (" + seed + "): divergence in keys between t and m (polymorphic method)");
    System.exit(1);
   }

#if (KEYS_REFERENCE) && ! (VALUES_REFERENCE)
   if ((m.GET_VALUE(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
    t.get(KEY2OBJ(T)) != null &&
    ! VALUE2OBJ(m.GET_VALUE(T)).equals(t.get(KEY2OBJ(T))))
#else
    if
 ((m.get(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
     t.get(KEY2OBJ(T)) != null &&
     ! m.get(KEY2OBJ(T)).equals(t.get(KEY2OBJ(T))))
#endif
     {
      System.out.println("Error (" + seed + "): divergence between t and m (polymorphic method)");
      System.exit(1);
     }
  }

  /* Again, we check that inquiries about random data give the same answer in m and t, but
     for m we use the standard method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();
   if (!valEquals(m.get(KEY2OBJ(T)), t.get(KEY2OBJ(T)))) {
    System.out.println("Error (" + seed + "): divergence between t and m (standard method)");
    System.exit(1);
   }
  }

  /* Now we put and remove random data in m and t, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();
   VALUE_TYPE U = genValue();
   if (!valEquals(m.put(KEY2OBJ(T), VALUE2OBJ(U)), t.put(KEY2OBJ(T),
 VALUE2OBJ(U)))) {
    System.out.println("Error (" + seed + "): divergence in put() between t and m");
    System.exit(1);
```

```
 }
 T = genKey();
 if (!valEquals(m.remove(KEY2OBJ(T)), t.remove(KEY2OBJ(T)))) {
  System.out.println("Error (" + seed + "): divergence in remove() between t and m");
  System.exit(1);
 }
}

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after removal");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after removal");


/* Now we check that m actually holds the same data. */

for(java.util.Iterator i=t.entrySet().iterator(); i.hasNext();) {
 java.util.Map.Entry e = (java.util.Map.Entry)i.next();
 if (!valEquals(e.getValue(), m.get(e.getKey()))) {
  System.out.println("Error (" + seed + "): m and t differ on an entry ("+e+") after removal (iterating on t)");
  System.exit(1);
 }
}

/* Now we check that m actually holds that data, but iterating on m. */

for(java.util.Iterator
i=m.entrySet().iterator(); i.hasNext();) {
 java.util.Map.Entry e = (java.util.Map.Entry)i.next();
 if (!valEquals(e.getValue(), t.get(e.getKey()))) {
  System.out.println("Error (" + seed + "): m and t differ on an entry ("+e+") after removal (iterating on m)");
  System.exit(1);
 }
}

/* Now we check that m actually holds the same keys. */

for(java.util.Iterator i=t.keySet().iterator(); i.hasNext();) {
 Object o = i.next();
 if (!m.containsKey(o)) {
  System.out.println("Error (" + seed + "): m and t differ on a key ("+o+") after removal (iterating on t)");
  System.exit(1);
 }
 if (!m.keySet().contains(o)) {
  System.out.println("Error (" + seed + "): m and t differ on a key ("+o+", in keySet()) after removal (iterating on t)");
  System.exit(1);
 }
}

/* Now we check that m actually holds the same keys, but iterating on m. */
```

```
 for(java.util.Iterator i=m.keySet().iterator(); i.hasNext();) {
  Object o = i.next();
  if (!t.containsKey(o))
 {
   System.out.println("Error (" + seed + "): m and t differ on a key after removal (iterating on m)");
   System.exit(1);
  }
  if (!t.keySet().contains(o)) {
   System.out.println("Error (" + seed + "): m and t differ on a key (in keySet()) after removal (iterating on m)");
   System.exit(1);
  }
 }


 /* Now we check that m actually hold the same values. */

 for(java.util.Iterator i=t.values().iterator(); i.hasNext();) {
  Object o = i.next();
  if (!m.containsValue(o)) {
   System.out.println("Error (" + seed + "): m and t differ on a value after removal (iterating on t)");
   System.exit(1);
  }
  if (!m.values().contains(o)) {
   System.out.println("Error (" + seed + "): m and t differ on a value (in values()) after removal (iterating on t)");
   System.exit(1);
  }
 }

 /* Now we check that m actually hold the same values, but iterating on m. */

 for(java.util.Iterator i=m.values().iterator(); i.hasNext();) {
  Object o = i.next();
  if (!t.containsValue(o))
 {
   System.out.println("Error (" + seed + "): m and t differ on a value after removal (iterating on m)");
   System.exit(1);
  }
  if (!t.values().contains(o)) {
   System.out.println("Error (" + seed + "): m and t differ on a value (in values()) after removal (iterating on m)");
   System.exit(1);
  }
 }


 int h = m.hashCode();
```

```
  /* Now we save and read m. */

  try {
   java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
   java.io.OutputStream os = new java.io.FileOutputStream(ff);
   java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

   oos.writeObject(m);
   oos.close();

   java.io.InputStream is = new java.io.FileInputStream(ff);
   java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

   m = (OPEN_HASH_MAP)ois.readObject();
   ois.close();
   ff.delete();
  }
  catch(Exception e) {
   e.printStackTrace();
   System.exit(1);
  }


#if !KEY_CLASS_Reference && !VALUE_CLASS_Reference
  if
 (m.hashCode() != h) System.out.println("Error (" + seed + "): hashCode() changed after save/read");

  /* Now we check that m actually holds that data. */

  for(java.util.Iterator i=t.keySet().iterator(); i.hasNext();) {
   Object o = i.next();
   if (!valEquals(m.get(o),t.get(o))) {
    System.out.println("Error (" + seed + "): m and t differ on an entry after save/read");
    System.exit(1);
   }
  }
#else
  m.clear();
  m.putAll(t);
#endif

  /* Now we put and remove random data in m and t, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();
   VALUE_TYPE U = genValue();
   if (!valEquals(m.put(KEY2OBJ(T), VALUE2OBJ(U)), t.put(KEY2OBJ(T), VALUE2OBJ(U)))) {
    System.out.println("Error (" + seed + "): divergence in put() between t and m after save/read");
```

```
    System.exit(1);
   }
   T = genKey();
   if (!valEquals(m.remove(KEY2OBJ(T)), t.remove(KEY2OBJ(T)))) {
    System.out.println("Error (" + seed + "): divergence
 in remove() between t and m after save/read");
    System.exit(1);
   }
  }

  if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after post-save/read removal");
  if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after post-save/read removal");



#ifdef Linked


  /* Now we play with iterators. */

  {
   java.util.ListIterator i, j;
   Object J;
   Map.Entry E, F;
   i = (java.util.ListIterator)m.entrySet().iterator();
   j = new java.util.LinkedList(t.entrySet()).listIterator();

   for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + seed + "): divergence in hasNext()");
    ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + seed + "): divergence in hasPrevious()");

    if (r.nextFloat() < .8 && i.hasNext()) {
#ifdef Custom
     ensure(m.strategy().equals((E=(java.util.Map.Entry)i.next()).getKey(), J = (F=(Map.Entry)j.next()).getKey()), "Error
(" + seed + "): divergence in next()");
#else
     ensure((E=(java.util.Map.Entry)i.next()).getKey().equals(J
 = (F=(Map.Entry)j.next()).getKey()), "Error (" + seed + "): divergence in next()");
#endif

     if (r.nextFloat() < 0.3) {
      i.remove();
      j.remove();
      t.remove(J);
     }
     else if (r.nextFloat() < 0.3) {
      Object U = VALUE2OBJ(genValue());
      E.setValue(U);
```

```
      t.put(F.getKey(), U);
     }
    }
    else if (r.nextFloat() < .2 && i.hasPrevious()) {
#ifdef Custom
     ensure(m.strategy().equals((E=(java.util.Map.Entry)i.previous()).getKey(), J =
(F=(Map.Entry)j.previous()).getKey()), "Error (" + seed + "): divergence in previous()");
#else
     ensure((E=(java.util.Map.Entry)i.previous()).getKey().equals(J = (F=(Map.Entry)j.previous()).getKey()), "Error (" +
seed + "): divergence in previous()");
#endif

     if (r.nextFloat() < 0.3) {
      i.remove();
      j.remove();
      t.remove(J);
     }
     else if (r.nextFloat() < 0.3) {
      Object U = VALUE2OBJ(genValue());
      E.setValue(U);
      t.put(F.getKey(),
 U);
     }
    }

    ensure(i.nextIndex() == j.nextIndex(), "Error (" + seed + "): divergence in nextIndex()");
    ensure(i.previousIndex() == j.previousIndex(), "Error (" + seed + "): divergence in previousIndex()");

   }

  }

  if (t.size() > 0) {
   java.util.ListIterator i, j;
   Object J;
   j = new java.util.LinkedList(t.keySet()).listIterator();
   int e = r.nextInt(t.size());
   Object from;
   do from = j.next(); while(e-- != 0);

   i = (java.util.ListIterator)((SORTED_SET)m.keySet()).iterator(KEY_OBJ2TYPE(from));

   for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + seed + "): divergence in hasNext() (iterator with starting point " +
from + ")");
    ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + seed + "): divergence in hasPrevious() (iterator with starting
point " + from + ")");
```

```
    if (r.nextFloat() < .8 && i.hasNext()) {
#ifdef Custom
     ensure(m.strategy().equals(i.next(), J = j.next()), "Error (" + seed + "):
 divergence in next() (iterator with starting point " + from + ")");
#else
     ensure(i.next().equals(J = j.next()), "Error (" + seed + "): divergence in next() (iterator with starting point " + from +
")");
#endif

     if (r.nextFloat() < 0.5) {
      i.remove();
      j.remove();
      t.remove(J);
     }
    }
    else if (r.nextFloat() < .2 && i.hasPrevious()) {
#ifdef Custom
     ensure(m.strategy().equals(i.previous(), J = j.previous()), "Error (" + seed + "): divergence in previous() (iterator
with starting point " + from + ")");
#else
     ensure(i.previous().equals(J = j.previous()), "Error (" + seed + "): divergence in previous() (iterator with starting
point " + from + ")");
#endif

     if (r.nextFloat() < 0.5) {
      i.remove();
      j.remove();
      t.remove(J);
     }
    }

    ensure(i.nextIndex() == j.nextIndex(), "Error (" + seed + "): divergence in nextIndex() (iterator with starting point "
+ from + ")");
    ensure(i.previousIndex() == j.previousIndex(), "Error
 (" + seed + "): divergence in previousIndex() (iterator with starting point " + from + ")");

   }

  }



  /* Now we check that m actually holds that data. */

  ensure(m.equals(t), "Error (" + seed + "): ! m.equals(t) after iteration");
  ensure(t.equals(m), "Error (" + seed + "): ! t.equals(m) after iteration");

#endif
```

```
  /* Now we take out of m everything, and check that it is empty. */

  for(java.util.Iterator i=t.keySet().iterator(); i.hasNext();) m.remove(i.next());

  if (!m.isEmpty())  {
   System.out.println("Error (" + seed + "): m is not empty (as it should be)");
   System.exit(1);
  }

#ifdef NumericEnhancements
#if VALUE_CLASS_Byte || VALUE_CLASS_Character || VALUE_CLASS_Short || VALUE_CLASS_Integer ||
VALUE_CLASS_Long
  /* Now we check that increment works properly, using random data */

  {
   t.clear();
   m.clear();

   for(int k = 0; k < 2*n; k++) {
    KEY_TYPE T = genKey();
    VALUE_TYPE U = genValue();

    VALUE_TYPE rU = m.increment(T, U);
    VALUE_GENERIC_CLASS
 tU = (VALUE_GENERIC_CLASS) t.get(KEY2OBJ(T));
    if (null == tU) {
     ensure(m.defaultReturnValue() == rU, "Error (" + seed + "): map increment does not return proper starting value.");
     t.put(KEY2OBJ(T), VALUE2OBJ((VALUE_TYPE) (m.defaultReturnValue() + U)));
    }
    else {
     t.put(KEY2OBJ(T), VALUE2OBJ((VALUE_TYPE) (((VALUE_TYPE) tU) + U)));
    }
   }

   // Maps should contain identical values
   ensure(new java.util.HashMap(m).equals(new java.util.HashMap(t)),
    "Error(" + seed + "): incremented maps are not equal.");
  }
#endif
#endif


#if (KEY_CLASS_Integer || KEY_CLASS_Long) && (VALUE_CLASS_Integer || VALUE_CLASS_Long)
  m = new OPEN_HASH_MAP(n, f);
  t.clear();
  int x;
```

```
 /* Now we torture-test the hash table. This part is implemented only for integers and longs. */

 int p = m.key.length;

 for(int i=0; i<p; i++) {
  for (int j=0; j<20; j++) {
   m.put(i+(r.nextInt() % 10)*p, 1);
   m.remove(i+(r.nextInt() % 10)*p);
  }

  for (int j=-10; j<10;
j++) m.remove(i+j*p);
 }

 t.putAll(m);

 /* Now all table entries are REMOVED. */

 for(int i=0; i<(p*f)/10; i++) {
  for (int j=0; j<10; j++) {
   if (!valEquals(m.put(KEY2OBJ(x = i+(r.nextInt() % 10)*p), VALUE2OBJ(1)), t.put(KEY2OBJ(x),
VALUE2OBJ(1))))
    System.out.println("Error (" + seed + "): m and t differ on an entry during torture-test insertion.");
  }
 }

 if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after torture-test insertion");
 if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after torture-test insertion");

 for(int i=0; i<p/10; i++) {
  for (int j=0; j<10; j++) {
   if (!valEquals(m.remove(KEY2OBJ(x = i+(r.nextInt() % 10)*p)), t.remove(KEY2OBJ(x))))
    System.out.println("Error (" + seed + "): m and t differ on an entry during torture-test removal.");
  }
 }

 if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after torture-test removal");
 if (!t.equals(m)) System.out.println("Error
(" + seed + "): !t.equals(m) after torture-test removal");

 if (!m.equals(m.clone())) System.out.println("Error (" + seed + "): !m.equals(m.clone()) after torture-test removal");
 if (!((OPEN_HASH_MAP)m.clone()).equals(m)) System.out.println("Error (" + seed + "): !m.clone().equals(m)
after torture-test removal");

 m.trim();

 if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after trim()");
```

```
  if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after trim()");
#endif

  System.out.println("Test OK");
  return;

#endif
 }


 public static void main(String args[]) {
  float f = Hash.DEFAULT_LOAD_FACTOR;
  int n  = Integer.parseInt(args[1]);
  if (args.length>2) f = Float.parseFloat(args[2]);
  if (args.length > 3) r = new java.util.Random(seed = Long.parseLong(args[3]));

  try {
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, f, "speedComp".equals(args[0]));
   else if ("test".equals(args[0]))
 runTest(n, f);
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed: " + seed);
  }

 }

#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/OpenHashMap.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import VALUE_PACKAGE.VALUE_COLLECTION;
import VALUE_PACKAGE.VALUE_ABSTRACT_COLLECTION;
import VALUE_PACKAGE.VALUE_ITERATOR;
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;

#if KEYS_REFERENCE
import java.util.Comparator;
#endif

/** An abstract class providing basic methods for sorted
 maps implementing a type-specific interface. */

public abstract class ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC extends ABSTRACT_MAP
KEY_VALUE_GENERIC implements SORTED_MAP KEY_VALUE_GENERIC {

 private static final long serialVersionUID = -1773560792952436569L;

 protected ABSTRACT_SORTED_MAP() {}

 /** {@inheritDoc}
  *
  * <p>The view is backed by the sorted set returned by {@link java.util.Map#entrySet()}. Note that
  * <em>no attempt is made at caching the result of this method</em>, as this would
  * require adding some attributes that lightweight implementations would
  * not need. Subclasses may easily override this policy by calling
  * this method and caching the result, but implementors are encouraged to
  * write more efficient ad-hoc implementations.
  *
  * @return a sorted set view of the keys of this map; it may be safely cast to a type-specific interface.
  */
 @Override
 public SORTED_SET KEY_GENERIC keySet() {
  return new KeySet();
 }

 /** A wrapper exhibiting the
 keys of a map. */

 protected class KeySet extends ABSTRACT_SORTED_SET KEY_GENERIC {
  @Override
  public boolean contains(final KEY_TYPE k) { return containsKey(k); }
```

```
  @Override
  public int size() { return ABSTRACT_SORTED_MAP.this.size(); }

  @Override
  public void clear() { ABSTRACT_SORTED_MAP.this.clear(); }

  @Override
  public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return
ABSTRACT_SORTED_MAP.this.comparator(); }

  @Override
  public KEY_GENERIC_TYPE FIRST() { return FIRST_KEY(); }

  @Override
  public KEY_GENERIC_TYPE LAST() { return LAST_KEY(); }

  @Override
  public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) { return headMap(to).keySet(); }

  @Override
  public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) { return
tailMap(from).keySet(); }

  @Override
  public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_TYPE from, final KEY_GENERIC_TYPE
to) { return subMap(from, to).keySet(); }

  @Override
  public KEY_BIDI_ITERATOR KEY_GENERIC
 iterator(final KEY_GENERIC_TYPE from) { return new KeySetIterator
KEY_VALUE_GENERIC_DIAMOND(ENTRYSET().iterator(new BasicEntry
KEY_VALUE_GENERIC_DIAMOND(from, VALUE_NULL))); }

  @Override
  public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new KeySetIterator
KEY_VALUE_GENERIC_DIAMOND(SORTED_MAPS.fastIterator(ABSTRACT_SORTED_MAP.this)); }
 }

 /** A wrapper exhibiting a map iterator as an iterator on keys.
  *
  * <p>To provide an iterator on keys, just create an instance of this
  * class using the corresponding iterator on entries.
  */

 protected static class KeySetIterator KEY_VALUE_GENERIC implements KEY_BIDI_ITERATOR
KEY_GENERIC {
  protected final ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> i;
```

```
 public KeySetIterator(ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> i) {
  this.i = i;
 }

 @Override
 public KEY_GENERIC_TYPE NEXT_KEY() { return i.next().ENTRY_GET_KEY(); };

 @Override
 public KEY_GENERIC_TYPE PREV_KEY() { return i.previous().ENTRY_GET_KEY();
};

 @Override
 public boolean hasNext() { return i.hasNext(); }

 @Override
 public boolean hasPrevious() { return i.hasPrevious(); }
}

/** {@inheritDoc}
 *
 * <p>The view is backed by the sorted set returned by {@link java.util.Map#entrySet()}. Note that
 * <em>no attempt is made at caching the result of this method</em>, as this would
 * require adding some attributes that lightweight implementations would
 * not need. Subclasses may easily override this policy by calling
 * this method and caching the result, but implementors are encouraged to
 * write more efficient ad-hoc implementations.
 *
 * @return a type-specific collection view of the values contained in this map.
 */
@Override
public VALUE_COLLECTION VALUE_GENERIC values() {
 return new ValuesCollection();
}

/** A wrapper exhibiting the values of a map. */
protected class ValuesCollection extends VALUE_ABSTRACT_COLLECTION VALUE_GENERIC {
 @Override
 public VALUE_ITERATOR VALUE_GENERIC
iterator() { return new ValuesIterator
KEY_VALUE_GENERIC_DIAMOND(SORTED_MAPS.fastIterator(ABSTRACT_SORTED_MAP.this)); }

 @Override
 public boolean contains(final VALUE_TYPE k) { return containsValue(k); }

 @Override
 public int size() { return ABSTRACT_SORTED_MAP.this.size(); }
```

```
 @Override
 public void clear() { ABSTRACT_SORTED_MAP.this.clear(); }
}

/** A wrapper exhibiting a map iterator as an iterator on values.
 *
 * <p>To provide an iterator on values, just create an instance of this
 * class using the corresponding iterator on entries.
 */

protected static class ValuesIterator KEY_VALUE_GENERIC implements VALUE_ITERATOR
VALUE_GENERIC {
 protected final ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> i;

 public ValuesIterator(ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> i) {
  this.i = i;
 }

 @Override
 public VALUE_GENERIC_TYPE NEXT_VALUE() { return i.next().ENTRY_GET_VALUE(); };

 @Override
 public boolean hasNext() { return i.hasNext();
}
}
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractSortedMap.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package PACKAGE;

#if KEY_CLASS_Object
import java.util.Arrays;
import java.util.Comparator;

import it.unimi.dsi.fastutil.PriorityQueue;
#else
import java.util.Iterator;
#endif

import java.util.Collection;
import java.util.NoSuchElementException;


/** A type-specific heap-based priority queue.
*
*
 <p>Instances of this class represent a priority queue using a heap. The heap is enlarged as needed, but
* it is never shrunk. Use the {@link #trim()} method to reduce its size, if necessary.
*/

public class HEAP_PRIORITY_QUEUE KEY_GENERIC implements PRIORITY_QUEUE KEY_GENERIC,
java.io.Serializable {
 private static final long serialVersionUID = 1L;

 /** The heap array. */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 protected transient KEY_GENERIC_TYPE[] heap = KEY_GENERIC_ARRAY_CAST
ARRAYS.EMPTY_ARRAY;

 /** The number of elements in this queue. */
 protected int size;

 /** The type-specific comparator used in this queue. */
 protected KEY_COMPARATOR KEY_SUPER_GENERIC c;

 /** Creates a new empty queue with a given capacity and comparator.
  *
  * @param capacity the initial capacity of this queue.
  * @param c the comparator used in this queue, or {@code null} for the natural order.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public HEAP_PRIORITY_QUEUE(int capacity, KEY_COMPARATOR KEY_SUPER_GENERIC
 c) {
  if (capacity > 0) this.heap = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[capacity];
  this.c = c;
```

```
}

/** Creates a new empty queue with a given capacity and using the natural order.
 *
 * @param capacity the initial capacity of this queue.
 */
public HEAP_PRIORITY_QUEUE(int capacity) {
 this(capacity, null);
}

/** Creates a new empty queue with a given comparator.
 *
 * @param c the comparator used in this queue, or {@code null} for the natural order.
 */
public HEAP_PRIORITY_QUEUE(KEY_COMPARATOR KEY_SUPER_GENERIC c) {
 this(0, c);
}

/** Creates a new empty queue using the natural order.
 */
public HEAP_PRIORITY_QUEUE() {
 this(0, null);
}

/** Wraps a given array in a queue using a given comparator.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The first {@code size} element of the array will be rearranged so to form a heap (this is
 * more efficient than enqueing the elements of {@code a} one by one).
 *
 * @param
a an array.
 * @param size the number of elements to be included in the queue.
 * @param c the comparator used in this queue, or {@code null} for the natural order.
 */
 public HEAP_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a, int size, final KEY_COMPARATOR
KEY_SUPER_GENERIC c) {
 this(c);
 this.heap = a;
 this.size = size;
 HEAPS.makeHeap(a, size, c);
}


/** Wraps a given array in a queue using a given comparator.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The elements of the array will be rearranged so to form a heap (this is
```

```
 * more efficient than enqueing the elements of {@code a} one by one).
 *
 * @param a an array.
 * @param c the comparator used in this queue, or {@code null} for the natural order.
 */
 public HEAP_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a, final KEY_COMPARATOR
KEY_SUPER_GENERIC c) {
  this(a, a.length, c);
 }

 /** Wraps a given array in a queue using the natural order.
 *
 * <p>The queue returned by this
 method will be backed by the given array.
 * The first {@code size} element of the array will be rearranged so to form a heap (this is
 * more efficient than enqueing the elements of {@code a} one by one).
 *
 * @param a an array.
 * @param size the number of elements to be included in the queue.
 */
 public HEAP_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a, int size) {
  this(a, size, null);
 }


 /** Wraps a given array in a queue using the natural order.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The elements of the array will be rearranged so to form a heap (this is
 * more efficient than enqueing the elements of {@code a} one by one).
 *
 * @param a an array.
 */
 public HEAP_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a) {
  this(a, a.length);
 }

#if KEYS_PRIMITIVE

 /** Creates a queue using the elements in a type-specific collection using a given comparator.
 *
 * <p>This constructor is more efficient than enqueing the elements of
 {@code collection} one by one.
 *
 * @param collection a collection; its elements will be used to initialize the queue.
 * @param c the comparator used in this queue, or {@code null} for the natural order.
 */
 public HEAP_PRIORITY_QUEUE(final COLLECTION KEY_EXTENDS_GENERIC collection, final
```

```
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
 this(collection.TO_KEY_ARRAY(), c);
 }

 /** Creates a queue using the elements in a type-specific collection using the natural order.
  *
  * <p>This constructor is
  * more efficient than enqueing the elements of {@code collection} one by one.
  *
  * @param collection a collection; its elements will be used to initialize the queue.
  */
 public HEAP_PRIORITY_QUEUE(final COLLECTION KEY_EXTENDS_GENERIC collection) {
  this(collection, null);
 }

 /** Creates a queue using the elements in a collection using a given comparator.
  *
  * <p>This constructor is more efficient than enqueing the elements of {@code collection} one by one.
  *
  * @param collection
 a collection; its elements will be used to initialize the queue.
  * @param c the comparator used in this queue, or {@code null} for the natural order.
  */
 public HEAP_PRIORITY_QUEUE(final Collection<? extends KEY_GENERIC_CLASS> collection, final
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
 this(collection.size(), c);
 final Iterator<? extends KEY_GENERIC_CLASS> iterator = collection.iterator();
 final int size = collection.size();
 for(int i = 0 ; i < size; i++) heap[i] = KEY_OBJ2TYPE(iterator.next());
 }

 /** Creates a queue using the elements in a collection using the natural order.
  *
  * <p>This constructor is
  * more efficient than enqueing the elements of {@code collection} one by one.
  *
  * @param collection a collection; its elements will be used to initialize the queue.
  */
 public HEAP_PRIORITY_QUEUE(final Collection<? extends KEY_GENERIC_CLASS> collection) {
  this(collection, null);
 }
#else
 /** Creates a queue using the elements in a collection using a given comparator.

 *
  * <p>This constructor is more efficient than enqueing the elements of {@code collection} one by one.
  *
  * @param collection a collection; its elements will be used to initialize the queue.
```

```
 * @param c the comparator used in this queue, or {@code null} for the natural order.
 */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public HEAP_PRIORITY_QUEUE(final Collection<? extends KEY_GENERIC_CLASS> collection, final
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
  this(KEY_GENERIC_ARRAY_CAST collection.toArray(), c);
 }

 /** Creates a queue using the elements in a collection using the natural order.
  *
  * <p>This constructor is
  * more efficient than enqueing the elements of {@code collection} one by one.
  *
  * @param collection a collection; its elements will be used to initialize the queue.
  */
 public HEAP_PRIORITY_QUEUE(final Collection<? extends KEY_GENERIC_CLASS> collection) {
  this(collection, null);
 }
#endif

 @Override
  public void enqueue(KEY_GENERIC_TYPE x) {
  if (size
 == heap.length) heap = ARRAYS.grow(heap, size + 1);

  heap[size++] = x;
  HEAPS.upHeap(heap, size, size - 1, c);
 }

 @Override
 public KEY_GENERIC_TYPE DEQUEUE() {
  if (size == 0) throw new NoSuchElementException();

  final KEY_GENERIC_TYPE result = heap[0];
  heap[0] = heap[--size];
#if KEY_CLASS_Object
  heap[size] = null;
#endif
  if (size != 0) HEAPS.downHeap(heap, size, 0, c);
  return result;
 }

 @Override
 public KEY_GENERIC_TYPE FIRST() {
  if (size == 0) throw new NoSuchElementException();
  return heap[0];
 }
```

```
 @Override
 public void changed() {
  HEAPS.downHeap(heap, size, 0, c);
 }

 @Override
 public int size() { return size; }

 @Override
 public void clear() {
#if KEY_CLASS_Object
  Arrays.fill(heap, 0, size, null);
#endif
  size = 0;
 }

 /** Trims the underlying heap array so that it has exactly {@link #size()} elements. */

 public void trim() {
  heap = ARRAYS.trim(heap, size);
 }

 @Override
 public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return c; }

 private
 void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
  s.defaultWriteObject();
  s.writeInt(heap.length);
  for(int i = 0; i < size; i++) s.WRITE_KEY(heap[i]);
 }

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
  s.defaultReadObject();
  heap = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[s.readInt()];
  for(int i = 0; i < size; i++) heap[i] = KEY_GENERIC_CAST s.READ_KEY();
 }


#ifdef TEST

 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
```

```
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif KEY_CLASS_Object
  return Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }

 private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
 private
 static java.text.FieldPosition p = new java.text.FieldPosition(0);

 private static String format(double d) {
  StringBuffer s = new StringBuffer();
  return format.format(d, s, p).toString();
 }

 private static void speedTest(int n, boolean comp) {
  System.out.println("There are presently no speed tests for this class.");
 }


 private static void fatal(String msg) {
  System.out.println(msg);
  System.exit(1);
 }

 private static void ensure(boolean cond, String msg) {
  if (cond) return;
  fatal(msg);
 }

 private static boolean heapEqual(KEY_TYPE[] a, KEY_TYPE[] b, int sizea, int sizeb) {
  if (sizea != sizeb) return false;
  KEY_TYPE[] aa = (KEY_TYPE[])a.clone();
  KEY_TYPE[] bb = (KEY_TYPE[])b.clone();
  java.util.Arrays.sort(aa, 0, sizea);
  java.util.Arrays.sort(bb, 0, sizeb);
  while(sizea-- != 0) if (! KEY_EQUALS(aa[sizea], bb[sizea])) return false;
  return true;
 }

 private static KEY_TYPE k[];

 protected static void runTest(int n) {
  long
```

```
 ms;
  Exception mThrowsIllegal, tThrowsIllegal, mThrowsOutOfBounds, tThrowsOutOfBounds, mThrowsNoElement,
tThrowsNoElement;
  KEY_TYPE rm = KEY_NULL, rt = KEY_NULL;
  k = new KEY_TYPE[n];

  for(int i = 0; i < n; i++) k[i] = genKey();

  HEAP_PRIORITY_QUEUE m = new
HEAP_PRIORITY_QUEUE(COMPARATORS.NATURAL_COMPARATOR);
  ARRAY_PRIORITY_QUEUE t = new
ARRAY_PRIORITY_QUEUE(COMPARATORS.NATURAL_COMPARATOR);

  /* We add pairs to t. */
  for(int i = 0; i < n / 2;  i++) {
   t.enqueue(k[i]);
   m.enqueue(k[i]);
  }

  ensure(heapEqual(m.heap, t.array, m.size(), t.size()), "Error (" + seed + "): m and t differ after creation (" + m + ", "
+ t + ")");

  if (m.size() != 0) {
   ensure(KEY_EQUALS(m.FIRST(), t.FIRST()), "Error (" + seed + "): m and t differ in first element after creation ("
+ m.FIRST() + ", " + t.FIRST() + ")");
  }

  /* Now we add and remove random data in m and t, checking that the result is the same. */

  for(int i=0; i<2*n;  i++) {

   if (r.nextDouble() < 0.01) {
    t.clear();
    m.clear();
    for(int
 j = 0; j < n / 2;  j++) {
     t.enqueue(k[j]);
     m.enqueue(k[j]);
    }
   }

   KEY_TYPE T = genKey();

   mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

   try {
    m.enqueue(T);
```

```
  }
  catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
  catch (IllegalArgumentException e) { mThrowsIllegal = e; }

  try {
   t.enqueue(T);
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
  catch (IllegalArgumentException e) { tThrowsIllegal = e; }

  ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): enqueue()
divergence in IndexOutOfBoundsException for " + T + " (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds
+ ")");
  ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): enqueue() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal
 + ")");

  ensure(heapEqual(m.heap, t.array, m.size(), t.size()), "Error (" + seed + "): m and t differ after enqueue (" + m + ", "
+ t + ")");

  if (m.size() != 0) {
   ensure(KEY_EQUALS(m.FIRST(), t.FIRST()), "Error (" + seed + "): m and t differ in first element after enqueue ("
+ m.FIRST() + ", " + t.FIRST() + ")");
  }

  mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

  try {
   rm = m.DEQUEUE();
  }
  catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
  catch (IllegalArgumentException e) { mThrowsIllegal = e; }
  catch (NoSuchElementException e) { mThrowsNoElement = e; }

  try {
   rt = t.DEQUEUE();
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
  catch (IllegalArgumentException e) { tThrowsIllegal = e; }
  catch (NoSuchElementException e) { tThrowsNoElement = e; }

  ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error ("
 + seed + "): dequeue() divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
  ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): dequeue() divergence in
IllegalArgumentException  (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): dequeue() divergence
```

```
in NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
   if (mThrowsOutOfBounds == null) ensure(KEY_EQUALS(rt, rm) , "Error (" + seed + "): divergence in dequeue()
between t and m (" + rt + ", " + rm + ")");


   ensure(heapEqual(m.heap, t.array, m.size(), t.size()), "Error (" + seed + "): m and t differ after dequeue (" + m + ", "
+ t + ")");

   if (m.size() != 0) {
    ensure(KEY_EQUALS(m.FIRST(), t.FIRST()), "Error (" + seed + "): m and t differ in first element after dequeue ("
+ m.FIRST() + ", " + t.FIRST() + ")");
   }

   /* Now we save
 and read m. */

   try {
    java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
    java.io.OutputStream os = new java.io.FileOutputStream(ff);
    java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

    oos.writeObject(m);
    oos.close();

    java.io.InputStream is = new java.io.FileInputStream(ff);
    java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

    m = (HEAP_PRIORITY_QUEUE)ois.readObject();
    ois.close();
    ff.delete();
   }
   catch(Exception e) {
    e.printStackTrace();
    System.exit(1);
   }

   ensure(heapEqual(m.heap, t.array, m.size(), t.size()), "Error (" + seed + "): m and t differ after save/read");

   HEAP_PRIORITY_QUEUE m2 = new HEAP_PRIORITY_QUEUE(t.array, t.size());
   ARRAY_PRIORITY_QUEUE t2 = new ARRAY_PRIORITY_QUEUE(m.heap, m.size());
   m = m2;
   t = t2;

   ensure(heapEqual(m.heap, t.array, m.size(), t.size()), "Error (" + seed + "): m and t differ after wrap (" + m + ", " + t
+ ")");

   if (m.size()
```

```
 != 0) {
    ensure(KEY_EQUALS(m.FIRST(), t.FIRST()), "Error (" + seed + "): m and t differ in first element after wrap (" +
m.FIRST() + ", " + t.FIRST() + ")");
   }

   if (m.size() != 0 && ((new OPEN_HASH_SET(m.heap, 0, m.size)).size() == m.size())) {

    int j = t.size(), M = --j;
#if KEYS_PRIMITIVE
    while(j-- != 0) if (KEY_LESS(t.array[j], t.array[M])) M = j;
#else
    while(j-- != 0) if (((Comparable)t.array[j]).compareTo(t.array[M])< 0) M = j;
#endif

    m.heap[0] = t.array[M] = genKey();

    m.changed();
    t.changed();

    ensure(heapEqual(m.heap, t.array, m.size(), t.size()), "Error (" + seed + "): m and t differ after change (" + m + ", " +
t + ")");

    if (m.size() != 0) {
     ensure(KEY_EQUALS(m.FIRST(), t.FIRST()), "Error (" + seed + "): m and t differ in first element after change ("
+ m.FIRST() + ", " + t.FIRST() + ")");
    }
   }
  }


  /* Now we check that m actually holds the same data. */

  m.clear();
  ensure(m.isEmpty(), "Error (" + seed + "):
 m is not empty after clear()");

  System.out.println("Test OK");
 }



 public static void main(String args[]) {
  int n  = Integer.parseInt(args[1]);
  if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));


  try {
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
```

```
  else if ("test".equals(args[0])) runTest(n);
 } catch(Throwable e) {
  e.printStackTrace(System.err);
  System.err.println("seed: " + seed);
 }
}

#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/HeapPriorityQueue.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2019-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/test/it/unimi/dsi/fastutil/objects/Object2IntArrayMapTest.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

/**  An abstract class facilitating the creation of type-specific iterators.
*
* @deprecated As of fastutil 8 this class is no longer necessary, as its previous abstract
* methods are now default methods of the type-specific interface.
*/

@Deprecated
public abstract class KEY_ABSTRACT_ITERATOR KEY_GENERIC
 implements KEY_ITERATOR KEY_GENERIC {
 protected KEY_ABSTRACT_ITERATOR() {}
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractIterator.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.Arrays;
import java.util.Collection;
import java.util.Iterator;
import java.util.RandomAccess;
```

```
import java.util.NoSuchElementException;
#if KEYS_REFERENCE
import java.util.Comparator;
#endif

#if KEYS_PRIMITIVE

/** A type-specific array-based list; provides some additional methods that
 use polymorphism to avoid (un)boxing.
*
* <p>This class implements a lightweight, fast, open, optimized,
* reuse-oriented version of array-based lists. Instances of this class
* represent a list with an array that is enlarged as needed when new entries
* are created (by doubling its current length), but is
* <em>never</em> made smaller (even on a {@link #clear()}). A family of
* {@linkplain #trim() trimming methods} lets you control the size of the
* backing array; this is particularly useful if you reuse instances of this class.
* Range checks are equivalent to those of {@link java.util}'s classes, but
* they are delayed as much as possible. The backing array is exposed by the
* {@link #elements()} method.
*
* <p>This class implements the bulk methods {@code removeElements()},
* {@code addElements()} and {@code getElements()} using
* high-performance system calls (e.g., {@link
* System#arraycopy(Object,int,Object,int,int) System.arraycopy()} instead of
* expensive
 loops.
*
* @see java.util.ArrayList
*/

public class ARRAY_LIST KEY_GENERIC extends ABSTRACT_LIST KEY_GENERIC implements
RandomAccess, Cloneable, java.io.Serializable {
 private static final long serialVersionUID = -7046029254386353130L;


#else

/** A type-specific array-based list; provides some additional methods that use polymorphism to avoid (un)boxing.
*
* <p>This class implements a lightweight, fast, open, optimized,
* reuse-oriented version of array-based lists. Instances of this class
* represent a list with an array that is enlarged as needed when new entries
* are created (by doubling the current length), but is
* <em>never</em> made smaller (even on a {@link #clear()}). A family of
* {@linkplain #trim() trimming methods} lets you control the size of the
* backing array; this is particularly useful if you reuse instances of this class.
* Range checks are equivalent to those of {@link java.util}'s classes, but
```

```
* they are delayed as much as possible.
*
* <p>The
 backing array is exposed by the {@link #elements()} method. If an instance
* of this class was created {@linkplain #wrap(Object[],int) by wrapping},
* backing-array reallocations will be performed using reflection, so that
* {@link #elements()} can return an array of the same type of the original array: the comments
* about efficiency made in {@link it.unimi.dsi.fastutil.objects.ObjectArrays} apply here.
* Moreover, you must take into consideration that assignment to an array
* not of type {@code Object[]} is slower due to type checking.
*
* <p>This class implements the bulk methods {@code removeElements()},
* {@code addElements()} and {@code getElements()} using
* high-performance system calls (e.g., {@link
* System#arraycopy(Object,int,Object,int,int) System.arraycopy()} instead of
* expensive loops.
*
* @see java.util.ArrayList
*/

public class ARRAY_LIST KEY_GENERIC extends ABSTRACT_LIST KEY_GENERIC implements
RandomAccess, Cloneable, java.io.Serializable {
 private
 static final long serialVersionUID = -7046029254386353131L;


#endif

 /** The initial default capacity of an array list. */
 public static final int DEFAULT_INITIAL_CAPACITY = 10;

#if ! KEYS_PRIMITIVE
 /** Whether the backing array was passed to {@code wrap()}. In
  * this case, we must reallocate with the same type of array. */
 protected final boolean wrapped;
#endif

 /** The backing array. */
 protected transient KEY_GENERIC_TYPE a[];

 /** The current actual size of the list (never greater than the backing-array length). */
 protected int size;

 /** Creates a new array list using a given array.
  *
  * <p>This constructor is only meant to be used by the wrapping methods.
  *
  * @param a the array that will be used to back this array list.
```

```
  */

 protected ARRAY_LIST(final KEY_GENERIC_TYPE a[], @SuppressWarnings("unused") boolean dummy) {
  this.a = a;
#if ! KEYS_PRIMITIVE
  this.wrapped = true;
#endif
 }

 /** Creates a new array list with given capacity.
  *
  * @param capacity
 the initial capacity of the array list (may be 0).
  */

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public ARRAY_LIST(final int capacity) {
  if (capacity < 0) throw new IllegalArgumentException("Initial capacity (" + capacity + ") is negative");
  if (capacity == 0) a = KEY_GENERIC_ARRAY_CAST ARRAYS.EMPTY_ARRAY;
  else a = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[capacity];
#if ! KEYS_PRIMITIVE
  wrapped = false;
#endif
 }

 /** Creates a new array list with {@link #DEFAULT_INITIAL_CAPACITY} capacity. */

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public ARRAY_LIST() {
  a = KEY_GENERIC_ARRAY_CAST ARRAYS.DEFAULT_EMPTY_ARRAY; // We delay allocation
#if ! KEYS_PRIMITIVE
  wrapped = false;
#endif
 }

 /** Creates a new array list and fills it with a given collection.
  *
  * @param c a collection that will be used to fill the array list.
  */

 public ARRAY_LIST(final Collection<? extends KEY_GENERIC_CLASS> c) {
  this(c.size());
#if KEYS_PRIMITIVE
  size = ITERATORS.unwrap(ITERATORS.AS_KEY_ITERATOR(c.iterator()),
 a);
#else
  size = ITERATORS.unwrap(c.iterator(), a);
#endif
```

```
}

/** Creates a new array list and fills it with a given type-specific collection.
 *
 * @param c a type-specific collection that will be used to fill the array list.
 */

public ARRAY_LIST(final COLLECTION KEY_EXTENDS_GENERIC c) {
 this(c.size());
 size = ITERATORS.unwrap(c.iterator(), a);
}

/** Creates a new array list and fills it with a given type-specific list.
 *
 * @param l a type-specific list that will be used to fill the array list.
 */

public ARRAY_LIST(final LIST KEY_EXTENDS_GENERIC l) {
 this(l.size());
 l.getElements(0, a, 0, size = l.size());
}

/** Creates a new array list and fills it with the elements of a given array.
 *
 * @param a an array whose elements will be used to fill the array list.
 */

public ARRAY_LIST(final KEY_GENERIC_TYPE a[]) {
 this(a, 0, a.length);
}

/** Creates a new array list and fills it with the elements of a given array.
 *
 * @param a an array
whose elements will be used to fill the array list.
 * @param offset the first element to use.
 * @param length the number of elements to use.
 */

public ARRAY_LIST(final KEY_GENERIC_TYPE a[], final int offset, final int length) {
 this(length);
 System.arraycopy(a, offset, this.a, 0, length);
 size = length;
}

/** Creates a new array list and fills it with the elements returned by an iterator..
 *
 * @param i an iterator whose returned elements will fill the array list.
```

```
 */

 public ARRAY_LIST(final Iterator<? extends KEY_GENERIC_CLASS> i) {
  this();
  while(i.hasNext()) this.add(KEY_CLASS2TYPE(i.next()));
 }

 /** Creates a new array list and fills it with the elements returned by a type-specific iterator..
  *
  * @param i a type-specific iterator whose returned elements will fill the array list.
  */

 public ARRAY_LIST(final KEY_ITERATOR KEY_EXTENDS_GENERIC i) {
  this();
  while(i.hasNext()) this.add(i.NEXT_KEY());
 }

#if KEYS_PRIMITIVE
 /** Returns the backing
 array of this list.
  *
  * @return the backing array.
  */

 public KEY_GENERIC_TYPE[] elements() {
  return a;
 }
#else
 /** Returns the backing array of this list.
  *
  * <p>If this array list was created by wrapping a given array, it is guaranteed
  * that the type of the returned array will be the same. Otherwise, the returned
  * array will be of type {@link Object Object[]} (in spite of the declared return type).
  *
  * <p><strong>Warning</strong>: This behaviour may cause (unfathomable)
  * run-time errors if a method expects an array
  * actually of type {@code K[]}, but this methods returns an array
  * of type {@link Object Object[]}.
  *
  * @return the backing array.
  */

 public K[] elements() {
  return a;
 }
#endif

 /** Wraps a given array into an array list of given size.
```

```
 *
 * <p>Note it is guaranteed
 * that the type of the array returned by {@link #elements()} will be the same
 * (see the comments in the class documentation).
 *
 * @param a an array to wrap.

* @param length the length of the resulting array list.
 * @return a new array list of the given size, wrapping the given array.
 */

 public static KEY_GENERIC ARRAY_LIST KEY_GENERIC wrap(final KEY_GENERIC_TYPE a[], final int
length) {
  if (length > a.length) throw new IllegalArgumentException("The specified length (" + length + ") is greater than the
array size (" + a.length + ")");
  final ARRAY_LIST KEY_GENERIC l = new ARRAY_LIST KEY_GENERIC_DIAMOND(a, false);
  l.size = length;
  return l;
 }

 /** Wraps a given array into an array list.
 *
 * <p>Note it is guaranteed
 * that the type of the array returned by {@link #elements()} will be the same
 * (see the comments in the class documentation).
 *
 * @param a an array to wrap.
 * @return a new array list wrapping the given array.
 */

 public static KEY_GENERIC ARRAY_LIST KEY_GENERIC wrap(final KEY_GENERIC_TYPE a[]) {
  return wrap(a, a.length);
 }


 /** Ensures that this array list can contain the given number of
 entries without resizing.
 *
 * @param capacity the new minimum capacity for this array list.
 */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public void ensureCapacity(final int capacity) {
  if (capacity <= a.length || (a == ARRAYS.DEFAULT_EMPTY_ARRAY && capacity <=
DEFAULT_INITIAL_CAPACITY)) return;
#if KEYS_PRIMITIVE
  a = ARRAYS.ensureCapacity(a, capacity, size);
#else
  if (wrapped) a = ARRAYS.ensureCapacity(a, capacity, size);
```

```
  else {
   if (capacity > a.length) {
    final Object t[] = new Object[capacity];
    System.arraycopy(a, 0, t, 0, size);
    a = (KEY_GENERIC_TYPE[])t;
   }
  }
#endif
  assert size <= a.length;
 }

 /** Grows this array list, ensuring that it can contain the given number of entries without resizing,
  * and in case increasing the current capacity at least by a factor of 50%.
  *
  * @param capacity the new minimum capacity for this array list.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 private void grow(int capacity) {
  if (capacity <= a.length) return;
  if (a
 != ARRAYS.DEFAULT_EMPTY_ARRAY)
   capacity = (int)Math.max(Math.min((long)a.length + (a.length >> 1),
it.unimi.dsi.fastutil.Arrays.MAX_ARRAY_SIZE), capacity);
  else if (capacity < DEFAULT_INITIAL_CAPACITY) capacity = DEFAULT_INITIAL_CAPACITY;
#if KEYS_PRIMITIVE
  a = ARRAYS.forceCapacity(a, capacity, size);
#else
  if (wrapped) a = ARRAYS.forceCapacity(a, capacity, size);
  else {
   final Object t[] = new Object[capacity];
   System.arraycopy(a, 0, t, 0, size);
   a = (KEY_GENERIC_TYPE[])t;
  }
#endif
  assert size <= a.length;
 }

 @Override
 public void add(final int index, final KEY_GENERIC_TYPE k) {
  ensureIndex(index);
  grow(size + 1);
  if (index != size) System.arraycopy(a, index, a, index + 1, size - index);
  a[index] = k;
  size++;
  assert size <= a.length;
 }

 @Override
```

```
 public boolean add(final KEY_GENERIC_TYPE k) {
  grow(size + 1);
  a[size++] = k;
  assert size <= a.length;
  return true;
 }

 @Override
 public KEY_GENERIC_TYPE GET_KEY(final int index) {
  if (index
 >= size) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list size (" + size
+ ")");
  return a[index];
 }

 @Override
 public int indexOf(final KEY_TYPE k) {
  for(int i = 0; i < size; i++) if (KEY_EQUALS(k, a[i])) return i;
  return -1;
 }


 @Override
 public int lastIndexOf(final KEY_TYPE k) {
  for(int i = size; i-- != 0;) if (KEY_EQUALS(k, a[i])) return i;
  return -1;
 }

 @Override
 public KEY_GENERIC_TYPE REMOVE_KEY(final int index) {
  if (index >= size) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list size
(" + size + ")");
  final KEY_GENERIC_TYPE old = a[index];
  size--;
  if (index != size) System.arraycopy(a, index + 1, a, index, size - index);
#if KEYS_REFERENCE
  a[size] = null;
#endif
  assert size <= a.length;
  return old;
 }

 @Override
 public boolean REMOVE(final KEY_TYPE k) {
  int index = indexOf(k);
  if (index == -1) return false;
  REMOVE_KEY(index);
  assert size <= a.length;
```

```
 return
 true;
 }

 @Override
 public KEY_GENERIC_TYPE set(final int index, final KEY_GENERIC_TYPE k) {
  if (index >= size) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list size
(" + size + ")");
  KEY_GENERIC_TYPE old = a[index];
  a[index] = k;
  return old;
 }

 @Override
 public void clear() {
#if KEYS_REFERENCE
  Arrays.fill(a, 0, size, null);
#endif
  size = 0;
  assert size <= a.length;
 }

 @Override
 public int size() {
  return size;
 }

 @Override
 public void size(final int size) {
  if (size > a.length) a = ARRAYS.forceCapacity(a, size, this.size);
  if (size > this.size) Arrays.fill(a, this.size, size, KEY_NULL);
#if KEYS_REFERENCE
  else Arrays.fill(a, size, this.size, KEY_NULL);
#endif
  this.size = size;
 }

 @Override
 public boolean isEmpty() {
  return size == 0;
 }

 /** Trims this array list so that the capacity is equal to the size.
  *
  * @see java.util.ArrayList#trimToSize()
  */
 public void trim() {
  trim(0);
```

```
}

/**
Trims the backing array if it is too large.
 *
 * If the current array length is smaller than or equal to
 * {@code n}, this method does nothing. Otherwise, it trims the
 * array length to the maximum between {@code n} and {@link #size()}.
 *
 * <p>This method is useful when reusing lists.  {@linkplain #clear() Clearing a
 * list} leaves the array length untouched. If you are reusing a list
 * many times, you can call this method with a typical
 * size to avoid keeping around a very large array just
 * because of a few large transient lists.
 *
 * @param n the threshold for the trimming.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public void trim(final int n) {
 // TODO: use Arrays.trim() and preserve type only if necessary
 if (n >= a.length || size == a.length) return;
 final KEY_GENERIC_TYPE t[] = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[Math.max(n, size)];
 System.arraycopy(a, 0, t, 0, size);
 a = t;
 assert size <= a.length;
}


  /** Copies element of this
type-specific list into the given array using optimized system calls.
 *
 * @param from the start index (inclusive).
 * @param a the destination array.
 * @param offset the offset into the destination array where to store the first element copied.
 * @param length the number of elements to be copied.
 */
@Override
public void getElements(final int from, final KEY_TYPE[] a, final int offset, final int length) {
 ARRAYS.ensureOffsetLength(a, offset, length);
 System.arraycopy(this.a, from, a, offset, length);
}

/** Removes elements of this type-specific list using optimized system calls.
 *
 * @param from the start index (inclusive).
 * @param to the end index (exclusive).
 */
```

```
 @Override
 public void removeElements(final int from, final int to) {
  it.unimi.dsi.fastutil.Arrays.ensureFromTo(size, from, to);
  System.arraycopy(a, to, a, from, size - to);
  size -= (to - from);
#if KEYS_REFERENCE
  int i = to - from;
  while(i-- != 0) a[size + i] = null;
#endif
 }


 /** Adds
 elements to this type-specific list using optimized system calls.
  *
  * @param index the index at which to add elements.
  * @param a the array containing the elements.
  * @param offset the offset of the first element to add.
  * @param length the number of elements to add.
  */
 @Override
 public void addElements(final int index, final KEY_GENERIC_TYPE a[], final int offset, final int length) {
  ensureIndex(index);
  ARRAYS.ensureOffsetLength(a, offset, length);
  grow(size + length);
  System.arraycopy(this.a, index, this.a, index + length, size - index);
  System.arraycopy(a, offset, this.a, index, length);
  size += length;
 }

 /** Sets elements to this type-specific list using optimized system calls.
  *
  * @param index the index at which to start setting elements.
  * @param a the array containing the elements.
  * @param offset the offset of the first element to add.
  * @param length the number of elements to add.
  */
 @Override
 public void setElements(final int index,
 final KEY_GENERIC_TYPE a[], final int offset, final int length) {
  ensureIndex(index);
  ARRAYS.ensureOffsetLength(a, offset, length);
  if (index + length > size) throw new IndexOutOfBoundsException("End index (" + (index + length) + ") is greater
than list size (" + size + ")");
  System.arraycopy(a, offset, this.a, index, length);
 }

#if KEYS_PRIMITIVE
```

```
 @Override
 public KEY_TYPE[] toArray(KEY_TYPE a[]) {
  if (a == null || a.length < size) a = new KEY_TYPE[size];
  System.arraycopy(this.a, 0, a, 0, size);
  return a;
 }

 @Override
 public boolean addAll(int index, final COLLECTION c) {
  ensureIndex(index);
  int n = c.size();
  if (n == 0) return false;
  grow(size + n);
  if (index != size) System.arraycopy(a, index, a, index + n, size - index);
  final KEY_ITERATOR i = c.iterator();
  size += n;
  while(n-- != 0) a[index++] = i.NEXT_KEY();
  assert size <= a.length;
  return true;
 }

 @Override
 public boolean addAll(final int index, final LIST l) {
  ensureIndex(index);
  final
 int n = l.size();
  if (n == 0) return false;
  grow(size + n);
  if (index != size) System.arraycopy(a, index, a, index + n, size - index);
  l.getElements(0, a, index, n);
  size += n;
  assert size <= a.length;
  return true;
 }

 @Override
 public boolean removeAll(final COLLECTION c) {
  final KEY_TYPE[] a = this.a;
  int j = 0;
  for(int i = 0; i < size; i++)
   if (! c.contains(a[i])) a[j++] = a[i];
#if KEYS_REFERENCE
  Arrays.fill(a, j, size, null);
#endif
  final boolean modified = size != j;
  size = j;
  return modified;
```

```
 }

#endif

 @Override
 public boolean removeAll(final Collection<?> c) {
  final KEY_TYPE[] a = this.a;
  int j = 0;
  for(int i = 0; i < size; i++)
   if (! c.contains(KEY2OBJ(a[i]))) a[j++] = a[i];
#if KEYS_REFERENCE
  Arrays.fill(a, j, size, null);
#endif
  final boolean modified = size != j;
  size = j;
  return modified;
 }

 @Override
 public KEY_LIST_ITERATOR KEY_GENERIC listIterator(final int index) {
  ensureIndex(index);

  return new KEY_LIST_ITERATOR
 KEY_GENERIC() {
    int pos = index, last = -1;

    @Override
    public boolean hasNext() { return pos < size; }
    @Override
    public boolean hasPrevious() { return pos > 0; }
    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() { if (! hasNext()) throw new NoSuchElementException(); return
a[last = pos++]; }
    @Override
    public KEY_GENERIC_TYPE PREV_KEY() { if (! hasPrevious()) throw new NoSuchElementException(); return
a[last = --pos]; }
    @Override
    public int nextIndex() { return pos; }
    @Override
    public int previousIndex() { return pos - 1; }
    @Override
    public void add(KEY_GENERIC_TYPE k) {
     ARRAY_LIST.this.add(pos++, k);
     last = -1;
    }
    @Override
    public void set(KEY_GENERIC_TYPE k) {
     if (last == -1) throw new IllegalStateException();
```

```
    ARRAY_LIST.this.set(last, k);
   }
   @Override
   public void remove() {
    if (last == -1) throw new IllegalStateException();
    ARRAY_LIST.this.REMOVE_KEY(last);
    /*
 If the last operation was a next(), we are removing an element *before* us, and we must decrease pos
correspondingly. */
    if (last < pos) pos--;
    last = -1;
   }
  };
 }

 @Override
 public void sort(final KEY_COMPARATOR KEY_SUPER_GENERIC comp) {
  if (comp == null) {
   ARRAYS.stableSort(a, 0, size);
  } else {
   ARRAYS.stableSort(a, 0, size, comp);
  }
 }

 @Override
 public void unstableSort(final KEY_COMPARATOR KEY_SUPER_GENERIC comp) {
  if (comp == null) {
   ARRAYS.unstableSort(a, 0, size);
  } else {
   ARRAYS.unstableSort(a, 0, size, comp);
  }
 }

 @Override
 public ARRAY_LIST KEY_GENERIC clone() {
  ARRAY_LIST KEY_GENERIC c = new ARRAY_LIST KEY_GENERIC_DIAMOND(size);
  System.arraycopy(a, 0, c.a, 0, size);
  c.size = size;
  return c;
 }

#if KEY_CLASS_Object
 private boolean valEquals(final K a, final K b) { return a == null ? b == null : a.equals(b); }
#endif

 /** Compares this type-specific array list to another one.
  *
  * <p>This method exists only for sake
```

```
 of efficiency. The implementation
  * inherited from the abstract implementation would already work.
  *
  * @param l a type-specific array list.
  * @return true if the argument contains the same elements of this type-specific array list.
  */
 public boolean equals(final ARRAY_LIST KEY_GENERIC l) {
  if (l == this) return true;
  int s = size();
  if (s != l.size()) return false;
  final KEY_GENERIC_TYPE[] a1 = a;
  final KEY_GENERIC_TYPE[] a2 = l.a;

#if KEY_CLASS_Object
  while(s-- !=  0) if (! valEquals(a1[s], a2[s])) return false;
#else
  while(s-- !=  0) if (a1[s] != a2[s]) return false;
#endif
  return true;
 }


#if ! KEY_CLASS_Reference

 /** Compares this array list to another array list.
  *
  * <p>This method exists only for sake of efficiency. The implementation
  * inherited from the abstract implementation would already work.
  *
  * @param l an array list.
  * @return a negative integer,
  * zero, or a positive integer as this list is lexicographically less than, equal

 * to, or greater than the argument.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public int compareTo(final ARRAY_LIST KEY_EXTENDS_GENERIC l) {
  final int s1 = size(), s2 = l.size();
  final KEY_GENERIC_TYPE a1[] = a, a2[] = l.a;
  KEY_GENERIC_TYPE e1, e2;
  int r, i;

  for(i = 0; i < s1 && i < s2; i++) {
   e1 = a1[i];
   e2 = a2[i];
   if ((r = KEY_CMP(e1, e2)) != 0) return r;
  }
```

```
  return i < s2 ? -1 : (i < s1 ? 1 : 0);
 }
#endif


 private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
  s.defaultWriteObject();
  for(int i = 0; i < size; i++) s.WRITE_KEY(a[i]);
 }

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
  s.defaultReadObject();
  a = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[size];
  for(int i = 0; i < size; i++) a[i] = KEY_GENERIC_CAST s.READ_KEY();
 }


#ifdef TEST

 private static long seed = System.currentTimeMillis();
 private static java.util.Random
 r = new java.util.Random(seed);

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif KEY_CLASS_Object
  return Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }

 private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
 private static java.text.FieldPosition p = new java.text.FieldPosition(0);

 private static String format(double d) {
  StringBuffer s = new StringBuffer();
  return format.format(d, s, p).toString();
 }

 private static void speedTest(int n, boolean comp) {
  System.out.println("There are presently no speed tests for this class.");
 }
```

```
private static void fatal(String msg) {
 System.out.println(msg);
 System.exit(1);
}

private static void ensure(boolean cond, String msg) {
 if (cond) return;
 fatal(msg);
}

private static Object[] k, v, nk;
private static KEY_TYPE
kt[];
private static KEY_TYPE nkt[];
private static ARRAY_LIST topList;

protected static void testLists(LIST m, java.util.List t, int n, int level) {
 long ms;
 Exception mThrowsIllegal, tThrowsIllegal, mThrowsOutOfBounds, tThrowsOutOfBounds;
 Object rt = null;
 KEY_TYPE rm = KEY_NULL;

 if (level > 4) return;


 /* Now we check that both sets agree on random keys. For m we use the polymorphic method. */

 for(int i = 0; i < n; i++) {
  int p = r.nextInt() % (n * 2);

  KEY_TYPE T = genKey();
  mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

  try {
   m.set(p, T);
  }
  catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
  try {
   t.set(p, KEY2OBJ(T));
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

  ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
set() divergence at start in IndexOutOfBoundsException for index " + p + "  (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds
 + ")");
  if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(KEY2OBJ(m.GET_KEY(p))), "Error (" + level + ", " +
```

```
seed + "): m and t differ after set() on position " + p + " (" + m.GET_KEY(p) + ", " + t.get(p) + ")");

   p = r.nextInt() % (n * 2);
   mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

   try {
    m.GET_KEY(p);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   try {
    t.get(p);
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
get() divergence at start in IndexOutOfBoundsException for index " + p + "  (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
   if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(KEY2OBJ(m.GET_KEY(p))), "Error (" + level + ", " +
seed + "): m and t differ aftre get() on position " + p + " (" + m.GET_KEY(p) + ", " + t.get(p) + ")");

  }

  /* Now we check that
 both sets agree on random keys. For m we use the standard method. */

  for(int i = 0; i < n; i++) {
   int p = r.nextInt() % (n * 2);

   mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

   try {
    m.get(p);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   try {
    t.get(p);
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
get() divergence at start in IndexOutOfBoundsException for index " + p + "  (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
   if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(m.get(p)), "Error (" + level + ", " + seed + "): m and t
differ at start on position " + p + " (" + m.get(p) + ", " + t.get(p) + ")");

   }

  /* Now we check that m and t are equal. */
```

```
  if (!m.equals(t) || ! t.equals(m)) System.err.println("m: " + m + " t: " + t);

  ensure(m.equals(t), "Error (" + level + ",
 " + seed + "): ! m.equals(t) at start");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");



  /* Now we check that m actually holds that data. */
  for(Iterator i=t.iterator(); i.hasNext();) {
   ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */
  for(Iterator i=m.listIterator(); i.hasNext();) {
   ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
m)");
  }

  /* Now we check that inquiries about random data give the same answer in m and t. For
     m we use the polymorphic method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();
   ensure(m.contains(T) == t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in content
between t and m (polymorphic method)");
  }

  /* Again, we check that inquiries
 about random data give the same answer in m and t, but
     for m we use the standard method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();
   ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in
content between t and m (polymorphic method)");
  }

  /* Now we add and remove random data in m and t, checking that the result is the same. */

  for(int i=0; i<2*n;  i++) {
   KEY_TYPE T = genKey();

   try {
    m.add(T);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
```

```
  try {
   t.add(KEY2OBJ(T));
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

  T = genKey();
  int p = r.nextInt() % (2 * n + 1);

  mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

  try {
   m.add(p, T);
  }
  catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

  try {
   t.add(p, KEY2OBJ(T));
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }


  ensure((mThrowsOutOfBounds
 == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "): add() divergence in
IndexOutOfBoundsException for index " + p + " for " + T + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");

  p = r.nextInt() % (2 * n + 1);

  mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

  try {
   rm = m.REMOVE_KEY(p);
  }
  catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

  try {
   rt = t.remove(p);
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }


  ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
remove() divergence in IndexOutOfBoundsException for index " + p + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
  if (mThrowsOutOfBounds == null) ensure(rt.equals(KEY2OBJ(rm)), "Error (" + level + ", " + seed + "): divergence
in remove() between t and m (" + rt + ", " + rm + ")");
 }
```

```
  ensure(m.equals(t), "Error (" + level
 + ", " + seed + "): ! m.equals(t) after add/remove");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after add/remove");

  /* Now we add random data in m and t using addAll on a collection, checking that the result is the same. */

  for(int i=0; i<n;  i++) {
   int p = r.nextInt() % (2 * n + 1);
   Collection m1 = new java.util.ArrayList();
   int s = r.nextInt(n / 2 + 1);
   for(int j = 0; j < s; j++) m1.add(KEY2OBJ(genKey()));

   mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

   try {
    m.addAll(p, m1);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

   try {
    t.addAll(p, m1);
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
addAll() divergence in IndexOutOfBoundsException for index " + p + " for " + m1 + " (" + mThrowsOutOfBounds
+ ", " + tThrowsOutOfBounds + ")");

   ensure(m.equals(t),
 "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after addAll");
   ensure(t.equals(m), "Error (" + level + ", " + seed + m + t + "): ! t.equals(m) after addAll");
   }

  if (m.size() > n) {
   m.size(n);
   while(t.size() != n) t.remove(t.size() -1);
   }

  /* Now we add random data in m and t using addAll on a type-specific collection, checking that the result is the
same. */

  for(int i=0; i<n;  i++) {
   int p = r.nextInt() % (2 * n + 1);
   COLLECTION m1 = new ARRAY_LIST();
   Collection t1 = new java.util.ArrayList();
   int s = r.nextInt(n / 2 + 1);
   for(int j = 0; j < s; j++) {
```

```
  KEY_TYPE x = genKey();
  m1.add(x);
  t1.add(KEY2OBJ(x));
 }

 mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

 try {
  m.addAll(p, m1);
 }
 catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

 try {
  t.addAll(p, t1);
 }
 catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

 ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds
== null), "Error (" + level + ", " + seed + "): polymorphic addAll() divergence in IndexOutOfBoundsException for
index " + p + " for " + m1 + " (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");

 ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after polymorphic addAll");
 ensure(t.equals(m), "Error (" + level + ", " + seed + m + t + "): ! t.equals(m) after polymorphic addAll");
 }

 if (m.size() > n) {
  m.size(n);
  while(t.size() != n) t.remove(t.size() -1);
 }

 /* Now we add random data in m and t using addAll on a list, checking that the result is the same. */

 for(int i=0; i<n;  i++) {
  int p = r.nextInt() % (2 * n + 1);
  LIST m1 = new ARRAY_LIST();
  Collection t1 = new java.util.ArrayList();
  int s = r.nextInt(n / 2 + 1);
  for(int j = 0; j < s; j++) {
   KEY_TYPE x = genKey();
   m1.add(x);
   t1.add(KEY2OBJ(x));
  }

  mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

  try {
   m.addAll(p, m1);
  }
```

```
  catch
 (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

  try {
   t.addAll(p, t1);
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

  ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "): list
addAll() divergence in IndexOutOfBoundsException for index " + p + " for " + m1 + " (" + mThrowsOutOfBounds
+ ", " + tThrowsOutOfBounds + ")");

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after list addAll");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after list addAll");
 }

 /* Now we add random data in m and t using addElements, checking that the result is the same. */

 for(int i=0; i<n;  i++) {
  int p = r.nextInt() % (2 * n + 1);
  Collection t1 = new java.util.ArrayList();
  int s = r.nextInt(n / 2 + 1);
  KEY_TYPE a[] = new KEY_TYPE [s];
  for(int j = 0; j < s; j++) {
   KEY_TYPE x = genKey();
   t1.add(KEY2OBJ(x));
   a[j] = x;
  }

  mThrowsOutOfBounds
 = tThrowsOutOfBounds  = null;

  try {
   m.addElements(p, a);
  }
  catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

  try {
   t.addAll(p, t1);
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

  ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "): list
addElements() divergence in IndexOutOfBoundsException for index " + p + " for " + t1 + " (" +
mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after list addElements");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after list addElements");
```

```
  }

  if (m.size() > n) {
   m.size(n);
   while(t.size() != n) t.remove(t.size() -1);
  }

  /* Now we check that m actually holds the same data. */

  for(Iterator i=t.iterator(); i.hasNext();) {
   ensure(m.contains(i.next()), "Error (" + level + ", " + seed +
 "): m and t differ on an entry after removal (iterating on t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */

  for(Iterator i=m.listIterator(); i.hasNext();) {
   ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
m)");
  }

  /* Now we check that both sets agree on random keys. For m we use the standard method. */

  for(int i = 0; i < n; i++) {
   int p = r.nextInt() % (n * 2);

   mThrowsOutOfBounds = tThrowsOutOfBounds  = null;

   try {
    m.get(p);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   try {
    t.get(p);
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
get() divergence in IndexOutOfBoundsException for index " + p + "  (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
   if (mThrowsOutOfBounds ==
 null) ensure(t.get(p).equals(m.get(p)), "Error (" + level + ", " + seed + "): m and t differ on position " + p + " (" +
m.get(p) + ", " + t.get(p) +")");

  }

  /* Now we inquiry about the content with indexOf()/lastIndexOf(). */
```

```
  for(int i=0; i<10*n;  i++) {
   KEY_TYPE T = genKey();
   ensure(m.indexOf(KEY2OBJ(T)) == t.indexOf(KEY2OBJ(T)),
     "Error (" + level + ", " + seed + "): indexOf() divergence for " + T + "  (" + m.indexOf(KEY2OBJ(T)) + ", " +
t.indexOf(KEY2OBJ(T)) + ")");
   ensure(m.lastIndexOf(KEY2OBJ(T)) == t.lastIndexOf(KEY2OBJ(T)),
     "Error (" + level + ", " + seed + "): lastIndexOf() divergence for " + T + "  (" + m.lastIndexOf(KEY2OBJ(T)) + ", "
+ t.lastIndexOf(KEY2OBJ(T)) + ")");
   ensure(m.indexOf(T) == t.indexOf(KEY2OBJ(T)),
     "Error (" + level + ", " + seed + "): polymorphic indexOf() divergence for " + T + "  (" + m.indexOf(T) + ", " +
t.indexOf(KEY2OBJ(T)) + ")");
   ensure(m.lastIndexOf(T) == t.lastIndexOf(KEY2OBJ(T)),
     "Error (" + level + ", "
 + seed + "): polymorphic lastIndexOf() divergence for " + T + "  (" + m.lastIndexOf(T) + ", " +
t.lastIndexOf(KEY2OBJ(T)) + ")");
  }

  /* Now we check cloning. */

  if (level == 0) {
   ensure(m.equals(((ARRAY_LIST)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
   ensure(((ARRAY_LIST)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
  }

  /* Now we play with constructors. */
  ensure(m.equals(new ARRAY_LIST((Collection)m)), "Error (" + level + ", " + seed + "): m does not equal new
(Collection m)");
  ensure((new ARRAY_LIST((Collection)m)).equals(m), "Error (" + level + ", " + seed + "): new (Collection m)does
not equal m");
  ensure(m.equals(new ARRAY_LIST((COLLECTION)m)), "Error (" + level + ", " + seed + "): m does not equal
new (type-specific Collection m)");
  ensure((new ARRAY_LIST((COLLECTION)m)).equals(m), "Error (" + level + ", " + seed + "): new (type-specific
Collection m) does not equal m");
  ensure(m.equals(new
 ARRAY_LIST((LIST)m)), "Error (" + level + ", " + seed + "): m does not equal new (type-specific List m)");
  ensure((new ARRAY_LIST((LIST)m)).equals(m), "Error (" + level + ", " + seed + "): new (type-specific List m)
does not equal m");
  ensure(m.equals(new ARRAY_LIST(m.listIterator())), "Error (" + level + ", " + seed + "): m does not equal new
(m.listIterator())");
  ensure((new ARRAY_LIST(m.listIterator())).equals(m), "Error (" + level + ", " + seed + "): new (m.listIterator())
does not equal m");
  ensure(m.equals(new ARRAY_LIST(m.iterator())), "Error (" + level + ", " + seed + "): m does not equal new
(m.type_specific_iterator())");
  ensure((new ARRAY_LIST(m.iterator())).equals(m), "Error (" + level + ", " + seed + "): new
(m.type_specific_iterator()) does not equal m");

  /* Now we play with conversion to array, wrapping and copying. */
  ensure(m.equals(new ARRAY_LIST(m.TO_KEY_ARRAY())), "Error (" + level + ", " + seed + "): m does not
```

```
equal new (toArray(m))");
  ensure((new
 ARRAY_LIST(m.TO_KEY_ARRAY())).equals(m), "Error (" + level + ", " + seed + "): new (toArray(m)) does not
equal m");
  ensure(m.equals(wrap(m.TO_KEY_ARRAY())), "Error (" + level + ", " + seed + "): m does not equal wrap
(toArray(m))");
  ensure((wrap(m.TO_KEY_ARRAY())).equals(m), "Error (" + level + ", " + seed + "): wrap (toArray(m)) does not
equal m");


  int h = m.hashCode();

  /* Now we save and read m. */

  LIST m2 = null;

  try {
   java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
   java.io.OutputStream os = new java.io.FileOutputStream(ff);
   java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

   oos.writeObject(m);
   oos.close();

   java.io.InputStream is = new java.io.FileInputStream(ff);
   java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

   m2 = (LIST)ois.readObject();
   ois.close();
   ff.delete();
  }
  catch(Exception e) {
   e.printStackTrace();
   System.exit(1);
  }

#if
 ! KEY_CLASS_Reference
  ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

  /* Now we check that m2 actually holds that data. */

  ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
  ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
  /* Now we take out of m everything, and check that it is empty. */

  for(Iterator i=t.iterator(); i.hasNext();) m2.remove(i.next());
```

```
  ensure(m2.isEmpty(), "Error (" + level + ", " + seed + "): m2 is not empty (as it should be)");
#endif

  /* Now we play with iterators. */

  {
   KEY_LIST_ITERATOR i;
   java.util.ListIterator j;
   Object J;
   i = m.listIterator();
   j = t.listIterator();

   for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext()");
    ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level
 + ", " + seed + "): divergence in hasPrevious()");

    if (r.nextFloat() < .8 && i.hasNext()) {
     ensure(i.next().equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next()");

     if (r.nextFloat() < 0.2) {
      i.remove();
      j.remove();
     }
     else if (r.nextFloat() < 0.2) {
      KEY_TYPE T = genKey();
      i.set(T);
      j.set(KEY2OBJ(T));
     }
     else if (r.nextFloat() < 0.2) {
      KEY_TYPE T = genKey();
      i.add(T);
      j.add(KEY2OBJ(T));
     }
    }
    else if (r.nextFloat() < .2 && i.hasPrevious()) {
     ensure(i.previous().equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous()");

     if (r.nextFloat() < 0.2) {
      i.remove();
      j.remove();
     }
     else if (r.nextFloat() < 0.2) {
      KEY_TYPE T = genKey();
      i.set(T);
      j.set(KEY2OBJ(T));
     }
     else if (r.nextFloat() < 0.2) {
```

```
     KEY_TYPE T = genKey();
     i.add(T);
     j.add(KEY2OBJ(T));
    }
   }

   ensure(i.nextIndex()
 == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
   ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

  }

 }

 {
  Object previous = null;
  Object I, J;
  int from = r.nextInt(m.size() +1);
  KEY_LIST_ITERATOR i;
  java.util.ListIterator j;
  i = m.listIterator(from);
  j = t.listIterator(from);

  for(int k = 0; k < 2*n; k++) {
   ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting
point " + from + ")");
   ensure(i.hasPrevious() == j.hasPrevious() , "Error (" + level + ", " + seed + "): divergence in hasPrevious() (iterator
with starting point " + from + ")");

   if (r.nextFloat() < .8 && i.hasNext()) {
    ensure((I = i.next()).equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next() (" + I + ", " + J + ",
iterator with starting point " + from + ")");
    //System.err.println("Done
 next " + I + " " + J + "  " + badPrevious);

    if (r.nextFloat() < 0.2) {
     //System.err.println("Removing in next");
     i.remove();
     j.remove();
    }
    else if (r.nextFloat() < 0.2) {
     KEY_TYPE T = genKey();
     i.set(T);
     j.set(KEY2OBJ(T));
    }
    else if (r.nextFloat() < 0.2) {
     KEY_TYPE T = genKey();
     i.add(T);
```

```
     j.add(KEY2OBJ(T));
     }
    }
    else if (r.nextFloat() < .2 && i.hasPrevious()) {
     ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

     if (r.nextFloat() < 0.2) {
      //System.err.println("Removing in prev");
      i.remove();
      j.remove();
     }
     else if (r.nextFloat() < 0.2) {
      KEY_TYPE T = genKey();
      i.set(T);
      j.set(KEY2OBJ(T));
     }
     else if (r.nextFloat() < 0.2) {
      KEY_TYPE T = genKey();
      i.add(T);
      j.add(KEY2OBJ(T));
     }
    }
   }

  }

  /*
 Now we check that m actually holds that data. */

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after iteration");

  /* Now we select a pair of keys and create a subset. */

  if (! m.isEmpty()) {
   int start = r.nextInt(m.size());
   int end = start + r.nextInt(m.size() - start);
   //System.err.println("Checking subList from " + start + " to " + end + " (level=" + (level+1) + ")...");
   testLists(m.subList(start, end), t.subList(start, end), n, level + 1);

   ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after subList");
   ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subList");

  }

  m.clear();
  t.clear();
```

```
 ensure(m.isEmpty(), "Error (" + level + ", " + seed + "): m is not empty after clear()");
 }


 protected static void runTest(int n)
 {
  ARRAY_LIST m = new ARRAY_LIST();
  java.util.ArrayList t = new java.util.ArrayList();
  topList = m;
  k = new Object[n];
  nk = new Object[n];
  kt = new KEY_TYPE[n];
  nkt = new KEY_TYPE[n];

  for(int i = 0; i < n; i++) {
#if KEYS_REFERENCE
   k[i] = kt[i] = genKey();
   nk[i] = nkt[i] = genKey();
#else
   k[i] = new KEY_CLASS(kt[i] = genKey());
   nk[i] = new KEY_CLASS(nkt[i] = genKey());
#endif
  }

  /* We add pairs to t. */
  for(int i = 0; i < n;  i++) t.add(k[i]);

  /* We add to m the same data */
  m.addAll(t);

  testLists(m, t, n, 0);

  System.out.println("Test OK");
  return;
 }


 public static void main(String args[]) {
  int n  = Integer.parseInt(args[1]);
  if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));


  try {
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
   else if ("test".equals(args[0])) runTest(n);
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed:
```

```
" + seed);
  }
 }

#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/ArrayList.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

/** An abstract class providing basic methods for implementing a type-specific stack interface.
*
* @deprecated As of fastutil 8 this class is no longer necessary, as its previous abstract
* methods are now default methods of the type-specific interface.
*/
@Deprecated
public abstract class ABSTRACT_STACK
 KEY_GENERIC extends it.unimi.dsi.fastutil.AbstractStack<KEY_GENERIC_CLASS> implements STACK
KEY_GENERIC {
 protected ABSTRACT_STACK() {}
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractStack.drv

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.Collection;

import it.unimi.dsi.fastutil.objects.ObjectArrays;


/** A class providing static methods and objects that do useful things with type-specific collections.
*
* @see java.util.Collections
*/

public final class COLLECTIONS {

 private COLLECTIONS() {}

 /** An immutable
 class representing an empty type-specific collection.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific collection.
  */

 public abstract static class EmptyCollection KEY_GENERIC extends ABSTRACT_COLLECTION
KEY_GENERIC {
  protected EmptyCollection() {}

  @Override
  public boolean contains(KEY_TYPE k) { return false; }
```

```
  @Override
  public Object[] toArray() { return ObjectArrays.EMPTY_ARRAY; }

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
  public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return ITERATORS.EMPTY_ITERATOR; }

  @Override
  public int size() { return 0; }

  @Override
  public void clear() {}

  @Override
  public int hashCode() { return 0; }

  @Override
  public boolean equals(Object o) {
   if (o == this) return true;
   if (! (o instanceof Collection)) return false;
   return ((Collection<?>)o).isEmpty();
  }

  @Override
  public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException();
 }

  @Override
  public boolean removeAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean retainAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE

  @Override
  public boolean addAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean removeAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean retainAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

#endif
```

```
 }


 /** A synchronized wrapper class for collections. */

 public static class SynchronizedCollection KEY_GENERIC implements COLLECTION KEY_GENERIC,
java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final COLLECTION KEY_GENERIC collection;
  protected final Object sync;

  protected SynchronizedCollection(final COLLECTION KEY_GENERIC c, final
 Object sync) {
   if (c == null) throw new NullPointerException();
   this.collection = c;
   this.sync = sync;
  }

  protected SynchronizedCollection(final COLLECTION KEY_GENERIC c) {
   if (c == null) throw new NullPointerException();
   this.collection = c;
   this.sync = this;
  }

  @Override
  public boolean add(final KEY_GENERIC_TYPE k) { synchronized(sync) { return collection.add(k); } }

  @Override
  public boolean contains(final KEY_TYPE k) { synchronized(sync) { return collection.contains(k); } }

  @Override
  public boolean REMOVE(final KEY_TYPE k) { synchronized(sync) { return collection.REMOVE(k); } }

  @Override
  public int size() { synchronized(sync) { return collection.size(); } }

  @Override
  public boolean isEmpty() { synchronized(sync) { return collection.isEmpty(); } }

  @Override
  public KEY_TYPE[] TO_KEY_ARRAY() { synchronized(sync) { return collection.TO_KEY_ARRAY(); } }

#if KEYS_PRIMITIVE
  @Override
  public Object[] toArray() { synchronized(sync) {
```

```
 return collection.toArray(); } }

  /* {@inheritDoc}
   * @deprecated Please use {@code toArray()} instead&mdash;this method is redundant and will be removed in the
future.
   */
  @Deprecated
  @Override
  public KEY_TYPE[] TO_KEY_ARRAY(final KEY_TYPE[] a) { return toArray(a); }

  @Override
  public KEY_TYPE[] toArray(final KEY_TYPE[] a) { synchronized(sync) { return collection.toArray(a); } }

  @Override
  public boolean addAll(final COLLECTION c) { synchronized(sync) { return collection.addAll(c); } }

  @Override
  public boolean containsAll(final COLLECTION c) { synchronized(sync) { return collection.containsAll(c); } }

  @Override
  public boolean removeAll(final COLLECTION c) { synchronized(sync) { return collection.removeAll(c); } }

#ifdef JDK_PRIMITIVE_PREDICATE
  @Override
  public boolean removeIf(final JDK_PRIMITIVE_PREDICATE filter) { synchronized(sync) { return
collection.removeIf(filter); } }
#endif

  @Override
  public boolean retainAll(final COLLECTION c) { synchronized(sync)
 { return collection.retainAll(c); } }

  @Override
  @Deprecated
  public boolean add(final KEY_GENERIC_CLASS k) { synchronized(sync) { return collection.add(k); } }

  @Override
  @Deprecated
  public boolean contains(final Object k) { synchronized(sync) { return collection.contains(k); } }

  @Override
  @Deprecated
  public boolean remove(final Object k) { synchronized(sync) { return collection.remove(k); } }
#endif

  @Override
  public <T> T[] toArray(final T[] a) { synchronized(sync) { return collection.toArray(a); } }
```

```
  @Override
  public KEY_ITERATOR KEY_GENERIC iterator() { return collection.iterator(); }

  @Override
  public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) { synchronized(sync) { return
collection.addAll(c); } }

  @Override
  public boolean containsAll(final Collection<?> c) { synchronized(sync) { return collection.containsAll(c); } }

  @Override
  public boolean removeAll(final Collection<?> c) { synchronized(sync) { return collection.removeAll(c);
 } }

  @Override
  public boolean retainAll(final Collection<?> c) { synchronized(sync) { return collection.retainAll(c); } }

  @Override
  public void clear() { synchronized(sync) { collection.clear(); } }

  @Override
  public String toString() { synchronized(sync) { return collection.toString(); } }

  @Override
  public int hashCode() { synchronized(sync) { return collection.hashCode(); } }

  @Override
  public boolean equals(final Object o) { if (o == this) return true; synchronized(sync) { return collection.equals(o); }
}

  private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
   synchronized(sync) { s.defaultWriteObject(); }
  }
 }


 /** Returns a synchronized collection backed by the specified collection.
  *
  * @param c the collection to be wrapped in a synchronized collection.
  * @return a synchronized view of the specified collection.
  * @see java.util.Collections#synchronizedCollection(Collection)
  */
 public static
 KEY_GENERIC COLLECTION KEY_GENERIC synchronize(final COLLECTION KEY_GENERIC c) { return
new SynchronizedCollection KEY_GENERIC_DIAMOND(c); }

 /** Returns a synchronized collection backed by the specified collection, using an assigned object to synchronize.
  *
```

```
 * @param c the collection to be wrapped in a synchronized collection.
 * @param sync an object that will be used to synchronize the list access.
 * @return a synchronized view of the specified collection.
 * @see java.util.Collections#synchronizedCollection(Collection)
 */

 public static KEY_GENERIC COLLECTION KEY_GENERIC synchronize(final COLLECTION KEY_GENERIC
c, final Object sync) { return new SynchronizedCollection KEY_GENERIC_DIAMOND(c, sync); }


 /** An unmodifiable wrapper class for collections. */

 public static class UnmodifiableCollection KEY_GENERIC implements COLLECTION KEY_GENERIC,
java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final COLLECTION
 KEY_GENERIC collection;

  protected UnmodifiableCollection(final COLLECTION KEY_GENERIC c) {
   if (c == null) throw new NullPointerException();
   this.collection = c;
  }

  @Override
  public boolean add(final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException(); }

  @Override
  public boolean REMOVE(final KEY_TYPE k) { throw new UnsupportedOperationException(); }

  @Override
  public int size() { return collection.size(); }

  @Override
  public boolean isEmpty() { return collection.isEmpty(); }

  @Override
  public boolean contains(final KEY_TYPE o) { return collection.contains(o); }

  @Override
  public KEY_ITERATOR KEY_GENERIC iterator() { return ITERATORS.unmodifiable(collection.iterator()); }

  @Override
  public void clear() { throw new UnsupportedOperationException(); }

  @Override
  public <T> T[] toArray(final T[] a) { return collection.toArray(a); }
```

```
  @Override
  public Object[] toArray() { return collection.toArray(); }

  @Override
  public boolean containsAll(Collection<?>
 c) { return collection.containsAll(c); }

  @Override
  public boolean addAll(Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

  @Override
  public boolean removeAll(Collection<?> c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean retainAll(Collection<?> c) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
  @Override
  @Deprecated
  public boolean add(final KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException(); }

  @Override
  @Deprecated
  public boolean contains(final Object k) { return collection.contains(k); }

  @Override
  @Deprecated
  public boolean remove(final Object k) { throw new UnsupportedOperationException(); }

  @Override
  public KEY_TYPE[] TO_KEY_ARRAY() { return collection.TO_KEY_ARRAY(); }

  /* {@inheritDoc}
   * @deprecated Please use {@code toArray()} instead&mdash;this method is redundant.
   */
  @Deprecated
  @Override
  public KEY_TYPE[]
 TO_KEY_ARRAY(final KEY_TYPE[] a) { return toArray(a); }

  @Override
  public KEY_TYPE[] toArray(final KEY_TYPE[] a) { return collection.toArray(a); }

  @Override
  public boolean containsAll(COLLECTION c) { return collection.containsAll(c); }
```

```
  @Override
  public boolean addAll(COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean removeAll(COLLECTION c) { throw new UnsupportedOperationException(); }

  @Override
  public boolean retainAll(COLLECTION c) { throw new UnsupportedOperationException(); }
#endif
  @Override
  public String toString() { return collection.toString(); }

  @Override
  public int hashCode() { return collection.hashCode(); }

  @Override
  public boolean equals(final Object o) { if (o == this) return true; return collection.equals(o); }
 }


 /** Returns an unmodifiable collection backed by the specified collection.
  *
  * @param c the collection to be wrapped in an unmodifiable collection.
  * @return an unmodifiable view
 of the specified collection.
  * @see java.util.Collections#unmodifiableCollection(Collection)
  */
 public static KEY_GENERIC COLLECTION KEY_GENERIC unmodifiable(final COLLECTION
KEY_GENERIC c) { return new UnmodifiableCollection KEY_GENERIC_DIAMOND(c); }

 /** A collection wrapper class for iterables. */

 public static class IterableCollection KEY_GENERIC extends ABSTRACT_COLLECTION KEY_GENERIC
implements java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final KEY_ITERABLE KEY_GENERIC iterable;

  protected IterableCollection(final KEY_ITERABLE KEY_GENERIC iterable) {
   if (iterable == null) throw new NullPointerException();
   this.iterable = iterable;
  }

  @Override
  public int size() {
   int c = 0;
   final KEY_ITERATOR KEY_GENERIC iterator = iterator();
```

```
  while(iterator.hasNext()) {
   iterator.NEXT_KEY();
   c++;
  }

  return c;
 }

 @Override
 public boolean isEmpty() { return ! iterable.iterator().hasNext();
}

 @Override
 public KEY_ITERATOR KEY_GENERIC iterator() { return iterable.iterator(); }
}


/** Returns an unmodifiable collection backed by the specified iterable.
 *
 * @param iterable the iterable object to be wrapped in an unmodifiable collection.
 * @return an unmodifiable collection view of the specified iterable.
 */
 public static KEY_GENERIC COLLECTION KEY_GENERIC asCollection(final KEY_ITERABLE
KEY_GENERIC iterable) {
  if (iterable instanceof COLLECTION) return (COLLECTION KEY_GENERIC)iterable;
  return new IterableCollection KEY_GENERIC_DIAMOND(iterable);
 }

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Collections.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
*/


package PACKAGE;

import java.lang.Iterable;
#if KEYS_PRIMITIVE
import java.util.Objects;
import java.util.function.Consumer;

/** A type-specific {@link Iterable} that strengthens that specification of {@link #iterator()} and {@link
#forEach(Consumer)}.
*
* <p>Note that whenever there exist a primitive consumer in
 {@link java.util.function} (e.g., {@link java.util.function.IntConsumer}),
* trying to access any version of {@link #forEach(Consumer)} using a lambda expression with untyped arguments
* will generate an ambiguous method error. This can be easily solved by specifying the type of the argument, as in
* <pre>
*    intIterable.forEach((int x) -&gt; { // Do something with x });
* </pre>
* <p>The same problem plagues, for example, {@link
java.util.PrimitiveIterator.OfInt#forEachRemaining(java.util.function.IntConsumer)}.
*
* <p><strong>Warning</strong>: Java will let you write &ldquo;colon&rdquo; {@code for} statements with
primitive-type
* loop variables; however, what is (unfortunately) really happening is that at each iteration an
* unboxing (and, in the case of {@code fastutil} type-specific data structures, a boxing) will be performed. Watch
out.
*
* @see Iterable
*/

#else

/** A type-specific {@link Iterable} that strengthens that specification of {@link #iterator()}.
*
 * @see Iterable
*/

#endif


public interface KEY_ITERABLE KEY_GENERIC extends Iterable<KEY_GENERIC_CLASS> {

 /** Returns a type-specific iterator.
  *
  * <p>Note that this specification strengthens the one given in {@link Iterable#iterator()}.
```

```
  *
  * @return a type-specific iterator.
  * @see Iterable#iterator()
  */

 @Override
 KEY_ITERATOR KEY_GENERIC iterator();


#if KEYS_PRIMITIVE
 /**
  * Performs the given action for each element of this type-specific {@link java.lang.Iterable}
  * until all elements have been processed or the action throws an
  * exception.
  * @param action the action to be performed for each element.
  * @see java.lang.Iterable#forEach(java.util.function.Consumer)
  * @since 8.0.0
  */
#ifdef JDK_PRIMITIVE_KEY_CONSUMER
 @SuppressWarnings("overloads")
 default void forEach(final JDK_PRIMITIVE_KEY_CONSUMER action) {
#else
 default void forEach(final KEY_CONSUMER action) {
#endif
  Objects.requireNonNull(action);
  for(final KEY_ITERATOR iterator = iterator();
 iterator.hasNext();)
   action.accept(iterator.NEXT_KEY());
 }

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default void forEach(final Consumer<? super KEY_GENERIC_CLASS> action) {
#ifdef JDK_PRIMITIVE_KEY_CONSUMER
#ifdef KEY_WIDENED
  forEach(new JDK_PRIMITIVE_KEY_CONSUMER() {
   public void accept(final KEY_TYPE_WIDENED key) {
    action.accept(KEY2OBJ((KEY_TYPE)key));
   }
  });
#else
  forEach((JDK_PRIMITIVE_KEY_CONSUMER) action::accept);
#endif
#else
  forEach((KEY_CONSUMER) action::accept);
#endif
```

```
 }
#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Iterable.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.AbstractCollection;

/** An abstract class providing basic methods for collections implementing a type-specific interface.
*
* <p>In particular, this class provide {@link #iterator()}, {@code add()}, {@link #remove(Object)} and
* {@link #contains(Object)} methods that just call the
 type-specific counterpart.
*
* <p><strong>Warning</strong>: Because of a name clash between the list and collection interfaces
* the type-specific deletion method of a type-specific abstract
* collection is {@code rem()}, rather then {@code remove()}. A
* subclass must thus override {@code rem()}, rather than
* {@code remove()}, to make all inherited methods work properly.
*/

public abstract class ABSTRACT_COLLECTION KEY_GENERIC extends
AbstractCollection<KEY_GENERIC_CLASS> implements COLLECTION KEY_GENERIC {

 protected ABSTRACT_COLLECTION() {}
```

```
@Override
public abstract KEY_ITERATOR KEY_GENERIC iterator();

#if KEYS_PRIMITIVE

/** {@inheritDoc}
 *
 * <p>This implementation always throws an {@link UnsupportedOperationException}.
 */
@Override
public boolean add(final KEY_TYPE k) {
 throw new UnsupportedOperationException();
}

/** {@inheritDoc}
 *
 * <p>This implementation iterates over the elements in the collection,
 * looking for the specified element.
 */
@Override
public
boolean contains(final KEY_TYPE k) {
 final KEY_ITERATOR iterator = iterator();
 while (iterator.hasNext()) if (k == iterator.NEXT_KEY()) return true;
 return false;
}

/** {@inheritDoc}
 *
 * <p>This implementation iterates over the elements in the collection,
 * looking for the specified element and tries to remove it.
 */
@Override
public boolean rem(final KEY_TYPE k) {
 final KEY_ITERATOR KEY_GENERIC iterator = iterator();
 while (iterator.hasNext())
  if (k == iterator.NEXT_KEY()) {
   iterator.remove();
   return true;
  }
 return false;
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@SuppressWarnings("deprecation")
```

```
@Deprecated
@Override
public boolean add(final KEY_GENERIC_CLASS key) {
 return COLLECTION.super.add(key);
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public boolean contains(final Object
key) {
 return COLLECTION.super.contains(key);
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public boolean remove(final Object key) {
 return COLLECTION.super.remove(key);
}

@Override
public KEY_TYPE[] toArray(KEY_TYPE a[]) {
 if (a == null || a.length < size()) a = new KEY_TYPE[size()];
 ITERATORS.unwrap(iterator(), a);
 return a;
}

@Override
public KEY_TYPE[] TO_KEY_ARRAY() {
 return toArray((KEY_TYPE[]) null);
}

/** {@inheritDoc}
 * @deprecated Please use {@code toArray()} instead&mdash;this method is redundant and will be removed in the
future.
 */
@Deprecated
@Override
public KEY_TYPE[] TO_KEY_ARRAY(final KEY_TYPE a[]) {
 return toArray(a);
}

@Override
```

```
 public boolean addAll(final COLLECTION c) {
  boolean retVal = false;

  for(final KEY_ITERATOR i = c.iterator(); i.hasNext();)
   if (add(i.NEXT_KEY())) retVal = true;
  return retVal;
 }

 @Override
 public boolean containsAll(final
 COLLECTION c) {
  for(final KEY_ITERATOR i = c.iterator(); i.hasNext();)
   if (! contains(i.NEXT_KEY())) return false;
  return true;
 }

 @Override
 public boolean removeAll(final COLLECTION c) {
  boolean retVal = false;
  for(final KEY_ITERATOR i = c.iterator(); i.hasNext();)
   if (rem(i.NEXT_KEY())) retVal = true;
  return retVal;
 }

 @Override
 public boolean retainAll(final COLLECTION c) {
  boolean retVal = false;
  for(final KEY_ITERATOR i = iterator(); i.hasNext();)
   if (! c.contains(i.NEXT_KEY())) {
    i.remove();
    retVal = true;
   }
  return retVal;
 }
#endif

 @Override
 public String toString() {
  final StringBuilder s = new StringBuilder();
  final KEY_ITERATOR KEY_GENERIC i = iterator();
  int n = size();
  KEY_TYPE k;
  boolean first = true;

  s.append("{");

  while(n-- != 0) {
   if (first) first = false;
```

```
   else s.append(", ");
   k = i.NEXT_KEY();
#if KEYS_REFERENCE
   if (this == k) s.append("(this collection)"); else
#endif
    s.append(String.valueOf(k));
  }

  s.append("}");
  return
 s.toString();
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractCollection.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

#if KEY_CLASS_Object
import java.util.Arrays;
import java.util.Comparator;
import it.unimi.dsi.fastutil.PriorityQueue;
#endif

import java.util.NoSuchElementException;

/** A type-specific array-based priority queue.
*
```

```
* <p>Instances of this class represent a priority queue using a
 backing
* array&mdash;all operations are performed directly on the array. The array is
* enlarged as needed, but it is never shrunk. Use the {@link #trim()} method
* to reduce its size, if necessary.
*
* <p>This implementation is extremely inefficient, but it is difficult to beat
* when the size of the queue is very small.
*/

public class ARRAY_PRIORITY_QUEUE KEY_GENERIC implements PRIORITY_QUEUE KEY_GENERIC,
java.io.Serializable {
 private static final long serialVersionUID = 1L;

 /** The backing array. */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 protected transient KEY_GENERIC_TYPE array[] = KEY_GENERIC_ARRAY_CAST
ARRAYS.EMPTY_ARRAY;

 /** The number of elements in this queue. */
 protected int size;

 /** The type-specific comparator used in this queue. */
 protected KEY_COMPARATOR KEY_SUPER_GENERIC c;

 /** The first index, cached, if {@link #firstIndexValid} is true. */
 protected transient int firstIndex;

 /** Whether {@link #firstIndex} contains a valid value. */
 protected transient
 boolean firstIndexValid;

 /** Creates a new empty queue with a given capacity and comparator.
  *
  * @param capacity the initial capacity of this queue.
  * @param c the comparator used in this queue, or {@code null} for the natural order.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public ARRAY_PRIORITY_QUEUE(int capacity, KEY_COMPARATOR KEY_SUPER_GENERIC c) {
  if (capacity > 0) this.array = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[capacity];
  this.c = c;
 }

 /** Creates a new empty queue with a given capacity and using the natural order.
  *
  * @param capacity the initial capacity of this queue.
  */
 public ARRAY_PRIORITY_QUEUE(int capacity) {
```

```
 this(capacity, null);
 }

 /** Creates a new empty queue with a given comparator.
  *
  * @param c the comparator used in this queue, or {@code null} for the natural order.
  */
 public ARRAY_PRIORITY_QUEUE(KEY_COMPARATOR KEY_SUPER_GENERIC c) {
  this(0, c);
 }

 /** Creates a new empty queue using the natural order.
  */
 public ARRAY_PRIORITY_QUEUE()
 {
  this(0, null);
 }

 /** Wraps a given array in a queue using a given comparator.
  *
  * <p>The queue returned by this method will be backed by the given array.
  *
  * @param a an array.
  * @param size the number of elements to be included in the queue.
  * @param c the comparator used in this queue, or {@code null} for the natural order.
  */
 public ARRAY_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a, int size, final KEY_COMPARATOR
KEY_SUPER_GENERIC c) {
  this(c);
  this.array = a;
  this.size = size;
 }


 /** Wraps a given array in a queue using a given comparator.
  *
  * <p>The queue returned by this method will be backed by the given array.
  *
  * @param a an array.
  * @param c the comparator used in this queue, or {@code null} for the natural order.
  */
 public ARRAY_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a, final KEY_COMPARATOR
KEY_SUPER_GENERIC c) {
  this(a, a.length, c);
 }

 /** Wraps a given array in a queue using the natural order.
  *
```

```
 * <p>The
queue returned by this method will be backed by the given array.
 *
 * @param a an array.
 * @param size the number of elements to be included in the queue.
 */
public ARRAY_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a, int size) {
 this(a, size, null);
}


/** Wraps a given array in a queue using the natural order.
 *
 * <p>The queue returned by this method will be backed by the given array.
 *
 * @param a an array.
 */
public ARRAY_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a) {
 this(a, a.length);
}



/** Returns the index of the smallest element. */

SUPPRESS_WARNINGS_KEY_UNCHECKED
private int findFirst() {
 if (firstIndexValid) return this.firstIndex;
 firstIndexValid = true;
 int i = size;
 int firstIndex = --i;
 KEY_GENERIC_TYPE first = array[firstIndex];

 if (c == null) { while(i-- != 0) if (KEY_LESS(array[i], first)) first = array[firstIndex = i]; }
 else while(i-- != 0) { if (c.compare(array[i], first) < 0) first = array[firstIndex = i]; }

 return this.firstIndex
= firstIndex;
}

private void ensureNonEmpty() {
 if (size == 0) throw new NoSuchElementException();
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public void enqueue(KEY_GENERIC_TYPE x) {
 if (size == array.length) array = ARRAYS.grow(array, size + 1);
```

```
  if (firstIndexValid) {
   if (c == null) { if (KEY_LESS(x, array[firstIndex])) firstIndex = size; }
   else if (c.compare(x, array[firstIndex]) < 0) firstIndex = size;
  }
  else firstIndexValid = false;
  array[size++] = x;
 }

 @Override
 public KEY_GENERIC_TYPE DEQUEUE() {
  ensureNonEmpty();
  final int first = findFirst();
  final KEY_GENERIC_TYPE result = array[first];
  System.arraycopy(array, first + 1, array, first, --size - first);
#if KEY_CLASS_Object
  array[size] = null;
#endif
  firstIndexValid = false;
  return result;
 }

 @Override
 public KEY_GENERIC_TYPE FIRST() {
  ensureNonEmpty();
  return array[findFirst()];
 }

 @Override
 public void changed() {
  ensureNonEmpty();
  firstIndexValid = false;
 }

 @Override
 public
 int size() { return size; }

 @Override
 public void clear() {
#if KEY_CLASS_Object
  Arrays.fill(array, 0, size, null);
#endif
  size = 0;
  firstIndexValid = false;
 }

 /** Trims the underlying array so that it has exactly {@link #size()} elements. */
```

```
 public void trim() {
  array = ARRAYS.trim(array, size);
 }

 @Override
 public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return c; }

 private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
  s.defaultWriteObject();
  s.writeInt(array.length);
  for(int i = 0; i < size; i++) s.WRITE_KEY(array[i]);
 }

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
  s.defaultReadObject();
  array = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[s.readInt()];
  for(int i = 0; i < size; i++) array[i] = KEY_GENERIC_CAST s.READ_KEY();
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/ArrayPriorityQueue.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.Set;

/**  An abstract class providing basic methods for sets implementing a type-specific interface.
```

```
*
* <p>Note that the type-specific {@link Set} interface adds a type-specific {@code remove()}
* method, as it is no longer harmful for subclasses. Thus, concrete subclasses of this
 class
* must implement {@code remove()} (the {@code rem()} implementation of this
* class just delegates to {@code remove()}).
*/

public abstract class ABSTRACT_SET KEY_GENERIC extends ABSTRACT_COLLECTION KEY_GENERIC
implements Cloneable, SET KEY_GENERIC {

 protected ABSTRACT_SET() {}

 @Override
 public abstract KEY_ITERATOR KEY_GENERIC iterator();

 @Override
 public boolean equals(final Object o) {
  if (o == this) return true;
  if (!(o instanceof Set)) return false;

  Set<?> s = (Set<?>) o;
  if (s.size() != size()) return false;
  return containsAll(s);
 }


 /** Returns a hash code for this set.
  *
  * The hash code of a set is computed by summing the hash codes of
  * its elements.
  *
  * @return a hash code for this set.
  */
 @Override
 public int hashCode() {
  int h = 0, n = size();
  KEY_ITERATOR KEY_GENERIC i = iterator();
  KEY_GENERIC_TYPE k;

  while(n-- != 0) {
   k = i.NEXT_KEY(); // We need k because KEY2JAVAHASH() is a macro with repeated evaluation.
   h
 += KEY2JAVAHASH(k);
  }
  return h;
 }
```

```
#if KEYS_PRIMITIVE
 /** {@inheritDoc}
  * Delegates to the type-specific {@code rem()} method
  * implemented by type-specific abstract {@link java.util.Collection} superclass.
  */
 @Override
 public boolean remove(KEY_TYPE k) {
  return super.rem(k);
 }

 /** {@inheritDoc}
  * Delegates to the type-specific {@code remove()} method
  * specified in the type-specific {@link Set} interface.
  * @deprecated Please use {@code remove()} instead.
  */
 @Deprecated
 @Override
 public boolean rem(KEY_TYPE k) {
  return remove(k);
 }
#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractSet.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;
```

```
import static it.unimi.dsi.fastutil.BigArrays.copyFromBig;
import static it.unimi.dsi.fastutil.BigArrays.copyToBig;
import static it.unimi.dsi.fastutil.BigArrays.grow;
import static it.unimi.dsi.fastutil.BigArrays.trim;


import it.unimi.dsi.fastutil.BigArrays;
import it.unimi.dsi.fastutil.objects.AbstractObjectList;
import
 it.unimi.dsi.fastutil.objects.ObjectListIterator;
import it.unimi.dsi.fastutil.longs.LongArrays;

import java.io.Serializable;
import java.util.Iterator;
import java.util.Collection;
import java.util.NoSuchElementException;
import java.util.RandomAccess;

/** Compact storage of lists of arrays using front coding.
*
* <p>This class stores immutably a list of arrays in a single large array
* using front coding (of course, the compression will be reasonable only if
* the list is sorted lexicographically&mdash;see below). It implements an
* immutable type-specific list that returns the <var>i</var>-th array when
* calling {@link #get(int) get(<var>i</var>)}. The returned array may be
* freely modified.
*
* <p>Front coding is based on the idea that if the <var>i</var>-th and the
* (<var>i</var>+1)-th array have a common prefix, we might store the length
* of the common prefix, and then the rest of the second array.
*
* <p>This approach, of course, requires
 that once in a while an array is
* stored entirely.  The <em>ratio</em> of a front-coded list defines how
* often this happens (once every {@link #ratio()} arrays). A higher ratio
* means more compression, but means also a longer access time, as more arrays
* have to be probed to build the result. Note that we must build an array
* every time {@link #get(int)} is called, but this class provides also methods
* that extract one of the stored arrays in a given array, reducing garbage
* collection. See the documentation of the family of {@code get()}
* methods.
*
* <p>By setting the ratio to 1 we actually disable front coding: however, we
* still have a data structure storing large list of arrays with a reduced
* overhead (just one integer per array, plus the space required for lengths).
*
* <p>Note that the typical usage of front-coded lists is under the form of
```

```
* serialized objects; usually, the data that has to be compacted is processed
* offline, and the resulting structure
 is stored permanently. Since the
* pointer array is not stored, the serialized format is very small.
*
* <H2>Implementation Details</H2>
*
* <p>All arrays are stored in a {@linkplain it.unimi.dsi.fastutil.BigArrays big array}. A separate array of pointers
* indexes arrays whose position is a multiple of the ratio: thus, a higher ratio
* means also less pointers.
*
* <p>More in detail, an array whose position is a multiple of the ratio is
* stored as the array length, followed by the elements of the array. The array
* length is coded by a simple variable-length list of <var>k</var>-1 bit
* blocks, where <var>k</var> is the number of bits of the underlying primitive
* type.  All other arrays are stored as follows: let {@code common} the
* length of the maximum common prefix between the array and its predecessor.
* Then we store the array length decremented by {@code common}, followed
* by {@code common}, followed by the array elements whose index is
* greater than
 or equal to {@code common}. For instance, if we store
* {@code foo}, {@code foobar}, {@code football} and
* {@code fool} in a front-coded character-array list with ratio 3, the
* character array will contain
*
* <pre>
* <b>3</b> f o o <b>3</b> <b>3</b> b a r <b>5</b> <b>3</b> t b a l l <b>4</b> f o o l
* </pre>
*/

public class ARRAY_FRONT_CODED_LIST extends AbstractObjectList<KEY_TYPE[]> implements
Serializable, Cloneable, RandomAccess {

 private static final long serialVersionUID = 1L;

 /** The number of arrays in the list. */
 protected final int n;
 /** The ratio of this front-coded list. */
 protected final int ratio;
 /** The big array containing the compressed arrays. */
 protected final KEY_TYPE[][] array;
 /** The pointers to entire arrays in the list. */
 protected transient long[] p;

 /** Creates a new front-coded list containing the arrays returned by the given iterator.
  *
  * @param arrays an iterator returning arrays.
  * @param ratio the desired ratio.
```

```
*/

public ARRAY_FRONT_CODED_LIST(final Iterator<KEY_TYPE[]> arrays, final int ratio) {

 if (ratio < 1) throw new IllegalArgumentException("Illegal ratio (" + ratio + ")");

 KEY_TYPE[][] array = BIG_ARRAYS.EMPTY_BIG_ARRAY;
 long[] p = LongArrays.EMPTY_ARRAY;

 KEY_TYPE[][] a = new KEY_TYPE[2][];
 long curSize = 0;
 int n = 0, b = 0, common, length, minLength;

 while(arrays.hasNext()) {
  a[b] = arrays.next();
  length = a[b].length;

  if (n % ratio == 0) {
   p = LongArrays.grow(p, n / ratio + 1);
   p[n / ratio] = curSize;

   array = grow(array, curSize + count(length) + length, curSize);
   curSize += writeInt(array, length, curSize);
   copyToBig(a[b], 0, array, curSize, length);
   curSize += length;
  }
  else {
   minLength = a[1 - b].length;
   if (length < minLength) minLength = length;
   for(common = 0; common < minLength; common++) if (a[0][common] != a[1][common]) break;
   length -= common;

   array = grow(array, curSize + count(length) + count(common)
+ length, curSize);
   curSize += writeInt(array, length, curSize);
   curSize += writeInt(array, common, curSize);
   copyToBig(a[b], common, array, curSize, length);
   curSize += length;
  }

  b = 1 - b;
  n++;
 }

 this.n = n;
 this.ratio = ratio;
 this.array = trim(array, curSize);
```

```
 this.p = LongArrays.trim(p, (n + ratio - 1) / ratio);

}

/** Creates a new front-coded list containing the arrays in the given collection.
 *
 * @param c a collection containing arrays.
 * @param ratio the desired ratio.
 */

public ARRAY_FRONT_CODED_LIST(final Collection<KEY_TYPE[]> c, final int ratio) {
 this(c.iterator(), ratio);
}



/* The following (rather messy) methods implements the encoding of arbitrary integers inside a big array.
 * Unfortunately, we have to specify different codes for almost every type. */

/** Reads a coded length.
 * @param a the data big array.
 * @param pos the starting position.
 * @return the length coded at {@code pos}.
 */
private
static int readInt(final KEY_TYPE a[][], long pos) {
#if KEY_CLASS_Integer
 return BigArrays.get(a, pos);
#elif KEY_CLASS_Long
 return (int)BigArrays.get(a, pos);
#elif KEY_CLASS_Character
 final char c0 = BigArrays.get(a, pos);
 return c0  < 0x8000 ? c0 : (c0 & 0x7FFF) << 16 | BigArrays.get(a, pos + 1);
#elif KEY_CLASS_Short
 final short s0 = BigArrays.get(a, pos);
 return s0 >= 0 ? s0 : s0 << 16 | (BigArrays.get(a, pos + 1) & 0xFFFF);
#else
 final byte b0 = BigArrays.get(a, pos);
 if (b0  >= 0) return b0;
 final byte b1 = BigArrays.get(a, pos + 1);
 if (b1 >= 0) return (- b0 - 1) << 7 | b1;
 final byte b2 = BigArrays.get(a, pos + 2);
 if (b2 >= 0) return (- b0 - 1) << 14 | (- b1 - 1) << 7 | b2;
 final byte b3 = BigArrays.get(a, pos + 3);
 if (b3 >= 0) return (- b0 - 1) << 21 | (- b1 - 1) << 14 | (- b2 - 1) << 7 | b3;
 return (- b0 - 1) << 28 | (- b1 - 1) << 21 | (- b2 - 1) << 14 | (- b3 - 1) << 7 | BigArrays.get(a, pos + 4);
#endif
}
```

```
 /** Computes the number of
 elements coding a given length.
  * @param length the length to be coded.
  * @return the number of elements coding {@code length}.
  */
 private static int count(final int length) {
#if KEY_CLASS_Integer || KEY_CLASS_Long
  return 1;
#elif KEY_CLASS_Character || KEY_CLASS_Short
  return length < (1 << 15) ? 1 : 2;
#else
  if (length < (1 << 7)) return 1;
  if (length < (1 << 14)) return 2;
  if (length < (1 << 21)) return 3;
  if (length < (1 << 28)) return 4;
  return 5;
#endif
 }

 /** Writes a length.
  * @param a the data array.
  * @param length the length to be written.
  * @param pos the starting position.
  * @return the number of elements coding {@code length}.
  */
 private static int writeInt(final KEY_TYPE a[][], int length, long pos) {
#if KEY_CLASS_Long
  BigArrays.set(a, pos, length);
  return 1;
#elif KEY_CLASS_Integer
  BigArrays.set(a, pos, length);
  return 1;
#elif KEY_CLASS_Character
  if (length < (1 << 15)) {
   BigArrays.set(a, pos, (char)length);
   return 1;
  }
  BigArrays.set(a,
 pos++, (char)(length >>> 16 | 0x8000));
  BigArrays.set(a, pos, (char)(length & 0xFFFF));
  return 2;
#elif KEY_CLASS_Short
  if (length < (1 << 15)) {
   BigArrays.set(a, pos, (short)length);
   return 1;
  }
  BigArrays.set(a, pos++, (short)(- (length >>> 16) - 1));
```

```
  BigArrays.set(a, pos, (short)(length & 0xFFFF));
  return 2;
#else
  final int count = count(length);
  BigArrays.set(a, pos + count - 1, (byte)(length & 0x7F));

  if (count != 1) {
   int i = count - 1;
   while(i-- != 0) {
    length >>>= 7;
    BigArrays.set(a, pos + i, (byte)(- (length  & 0x7F) - 1));
   }
  }

  return count;
#endif
 }



 /** Returns the ratio of this list.
  *
  * @return the ratio of this list.
  */

 public int ratio() {
  return ratio;
 }


 /** Computes the length of the array at the given index.
  *
  * <p>This private version of {@link #arrayLength(int)} does not check its argument.
  *
  * @param index an index.
  * @return the length of the {@code index}-th array.

 */
 private int length(final int index) {
  final KEY_TYPE[][] array = this.array;
  final int delta = index % ratio; // The index into the p array, and the delta inside the block.

  long pos = p[index / ratio]; // The position into the array of the first entire word before the index-th.
  int length = readInt(array, pos);

  if (delta == 0) return length;

  // First of all, we recover the array length and the maximum amount of copied elements.
```

```
 int common;
 pos += count(length) + length;
 length = readInt(array, pos);
 common = readInt(array, pos + count(length));

 for(int i = 0; i < delta - 1; i++) {
  pos += count(length) + count(common) + length;
  length = readInt(array, pos);
  common = readInt(array, pos + count(length));
 }

 return length + common;
}


/** Computes the length of the array at the given index.
 *
 * @param index an index.
 * @return the length of the {@code index}-th array.
 */
public int arrayLength(final int index) {
 ensureRestrictedIndex(index);
 return
length(index);
}

/** Extracts the array at the given index.
 *
 * @param index an index.
 * @param a the array that will store the result (we assume that it can hold the result).
 * @param offset an offset into {@code a} where elements will be store.
 * @param length a maximum number of elements to store in {@code a}.
 * @return the length of the extracted array.
 */
private int extract(final int index, final KEY_TYPE a[], final int offset, final int length) {
 final int delta = index % ratio; // The delta inside the block.
 final long startPos = p[index / ratio]; // The position into the array of the first entire word before the index-th.
 long pos, prevArrayPos;
 int arrayLength = readInt(array, pos = startPos), currLen = 0, actualCommon;

 if (delta == 0) {
  pos = p[index / ratio] + count(arrayLength);
  copyFromBig(array, pos, a, offset, Math.min(length, arrayLength));
  return arrayLength;
 }

 int common = 0;
```

```
 for(int i = 0; i < delta;
i++) {
  prevArrayPos = pos + count(arrayLength) + (i != 0 ? count(common) : 0);
  pos = prevArrayPos + arrayLength;

  arrayLength = readInt(array, pos);
  common = readInt(array, pos + count(arrayLength));

  actualCommon = Math.min(common, length);
  if (actualCommon <= currLen) currLen = actualCommon;
  else {
   copyFromBig(array, prevArrayPos, a, currLen + offset, actualCommon - currLen);
   currLen = actualCommon;
  }
 }

 if (currLen < length) copyFromBig(array, pos + count(arrayLength) + count(common), a, currLen + offset,
Math.min(arrayLength, length - currLen));

 return arrayLength + common;
}

/** {@inheritDoc}
 * <p>This implementation delegates to {@link #getArray(int)}. */
@Override
public KEY_TYPE[] get(final int index) {
 return getArray(index);
}

/** Returns an array stored in this front-coded list.
 *
 * @param index an index.
 * @return the corresponding array stored in this front-coded list.
 */
public KEY_TYPE[] getArray(final int index)
{
 ensureRestrictedIndex(index);
 final int length = length(index);
 final KEY_TYPE a[] = new KEY_TYPE[length];
 extract(index, a, 0, length);
 return a;
}

/** Stores in the given array elements from an array stored in this front-coded list.
 *
 * @param index an index.
 * @param a the array that will store the result.
 * @param offset an offset into {@code a} where elements will be store.
```

```
 * @param length a maximum number of elements to store in {@code a}.
 * @return if {@code a} can hold the extracted elements, the number of extracted elements;
 * otherwise, the number of remaining elements with the sign changed.
 */
public int get(final int index, final KEY_TYPE[] a, final int offset, final int length) {
 ensureRestrictedIndex(index);
 ARRAYS.ensureOffsetLength(a, offset, length);

 final int arrayLength = extract(index, a, offset, length);
 if (length >= arrayLength) return arrayLength;
 return length - arrayLength;
}

/** Stores in the given array an array
stored in this front-coded list.
 *
 * @param index an index.
 * @param a the array that will store the content of the result (we assume that it can hold the result).
 * @return if {@code a} can hold the extracted elements, the number of extracted elements;
 * otherwise, the number of remaining elements with the sign changed.
 */
public int get(final int index, final KEY_TYPE[] a) {
 return get(index, a, 0, a.length);
}

@Override
public int size() {
 return n;
}

@Override
public ObjectListIterator<KEY_TYPE[]> listIterator(final int start) {
 ensureIndex(start);

 return new ObjectListIterator<KEY_TYPE[]>() {
   KEY_TYPE s[] = ARRAYS.EMPTY_ARRAY;
   int i = 0;
   long pos = 0;
   boolean inSync; // Whether the current value in a is the string just before the next to be produced.

   {
    if (start != 0) {
     if (start == n) i = start; // If we start at the end, we do nothing.
     else {
      pos = p[start / ratio];
      int j = start % ratio;
      i
```

```
= start - j;
      while(j-- != 0) next();
     }
    }
   }

   @Override
   public boolean hasNext() {
    return i < n;
   }

   @Override
   public boolean hasPrevious() {
    return i > 0;
   }

   @Override
   public int previousIndex() {
    return i - 1;
   }

   @Override
   public int nextIndex() {
    return i;
   }

   @Override
   public KEY_TYPE[] next() {
    int length, common;

    if (! hasNext()) throw new NoSuchElementException();

    if (i % ratio == 0) {
     pos = p[i / ratio];
     length = readInt(array, pos);
     s = ARRAYS.ensureCapacity(s, length, 0);
     copyFromBig(array, pos + count(length), s, 0, length);
     pos += length + count(length);
     inSync = true;
    }
    else {
     if (inSync) {
      length = readInt(array, pos);
      common = readInt(array, pos + count(length));
      s = ARRAYS.ensureCapacity(s, length + common, common);
      copyFromBig(array, pos + count(length) + count (common), s, common,
length);
      pos += count(length) + count(common) + length;
```

```
      length += common;
     }
     else {
      s = ARRAYS.ensureCapacity(s, length = length(i), 0);
      extract(i, s, 0, length);
     }
    }
    i++;
    return ARRAYS.copy(s, 0, length);
   }

   @Override
   public KEY_TYPE[] previous() {
    if (! hasPrevious()) throw new NoSuchElementException();
    inSync = false;
    return getArray(--i);
   }
  };
}


/** Returns a copy of this list.
 *
 *  @return a copy of this list.
 */
@Override
public ARRAY_FRONT_CODED_LIST clone() {
 return this;
}

@Override
public String toString() {
 final StringBuffer s = new StringBuffer();
 s.append("[");
 for(int i = 0; i < n; i++) {
  if (i != 0) s.append(", ");
  s.append(ARRAY_LIST.wrap(getArray(i)).toString());
 }
 s.append("]");
 return s.toString();
}

/** Computes the pointer array using the currently set ratio, number of elements and underlying array.
 *
 * @return the computed
pointer array.
 */
```

```
 protected long[] rebuildPointerArray() {
  final long[] p = new long[(n + ratio - 1) / ratio];
  final KEY_TYPE a[][] = array;
  int length, count;
  long pos = 0;

  for(int i = 0, j = 0, skip = ratio - 1; i < n; i++) {
   length = readInt(a, pos);
   count = count(length);
   if (++skip == ratio) {
    skip = 0;
    p[j++] = pos;
    pos += count + length;
   }
   else pos += count + count(readInt(a, pos + count)) + length;
  }

  return p;
 }


 private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
  s.defaultReadObject();

  // Rebuild pointer array
  p = rebuildPointerArray();
 }


#ifdef TEST

 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif KEY_CLASS_Object
  return
 Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }
```

```
 private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
 private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

 private static String format(double d) {
  StringBuffer s = new StringBuffer();
  return format.format(d, s, fp).toString();
 }

 private static void speedTest(int n, boolean comp) {
  System.out.println("There are presently no speed tests for this class.");
 }


 private static void fatal(String msg) {
  System.out.println(msg);
  System.exit(1);
 }

 private static void ensure(boolean cond, String msg) {
  if (cond) return;
  fatal(msg);
 }

 private static boolean contentEquals(java.util.List x, java.util.List y) {
  if (x.size() != y.size()) return false;
  for(int i = 0; i < x.size(); i++) if (! java.util.Arrays.equals((KEY_TYPE[])x.get(i), (KEY_TYPE[])y.get(i))) return
false;
  return true;
 }

 private static int
 l[];
 private static KEY_TYPE[][] a;


 private static void runTest(int n) {
  int c;

  l = new int[n];
  a = new KEY_TYPE[n][];

  for(int i = 0; i < n; i++) l[i] = (int)(Math.abs(r.nextGaussian())*32);
  for(int i = 0; i < n; i++) a[i] = new KEY_TYPE[l[i]];
  for(int i = 0; i < n; i++) for(int j = 0; j < l[i]; j++) a[i][j] = genKey();

  ARRAY_FRONT_CODED_LIST m = new
ARRAY_FRONT_CODED_LIST(it.unimi.dsi.fastutil.objects.ObjectIterators.wrap(a), r.nextInt(4) + 1);
  it.unimi.dsi.fastutil.objects.ObjectArrayList t = new it.unimi.dsi.fastutil.objects.ObjectArrayList(a);
```

```
 //System.out.println(m);
 //for(i = 0; i < t.size(); i++) System.out.println(ARRAY_LIST.wrap((KEY_TYPE[])t.get(i)));

 /* Now we check that m actually holds that data. */

 ensure(contentEquals(m, t), "Error (" + seed + "): m does not equal t at creation");

 /* Now we check cloning. */

 ensure(contentEquals(m, (java.util.List)m.clone()), "Error (" + seed + "): m does not equal m.clone()");

 /* Now we play
with iterators. */

 {
  ObjectListIterator i;
  java.util.ListIterator j;
  Object J;
  i = m.listIterator();
  j = t.listIterator();

  for(int k = 0; k < 2*n; k++) {
   ensure(i.hasNext() == j.hasNext(), "Error (" + seed + "): divergence in hasNext()");
   ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + seed + "): divergence in hasPrevious()");

   if (r.nextFloat() < .8 && i.hasNext()) {
    ensure(java.util.Arrays.equals((KEY_TYPE[])i.next(), (KEY_TYPE[])j.next()), "Error (" + seed + "): divergence in
next()");

   }
   else if (r.nextFloat() < .2 && i.hasPrevious()) {
    ensure(java.util.Arrays.equals((KEY_TYPE[])i.previous(), (KEY_TYPE[])j.previous()), "Error (" + seed + "):
divergence in previous()");
   }

   ensure(i.nextIndex() == j.nextIndex(), "Error (" + seed + "): divergence in nextIndex()");
   ensure(i.previousIndex() == j.previousIndex(), "Error (" + seed + "): divergence in previousIndex()");

  }

 }

 {
  Object previous = null;
  Object
I, J;
  int from = r.nextInt(m.size() +1);
```

```
  ObjectListIterator i;
  java.util.ListIterator j;
  i = m.listIterator(from);
  j = t.listIterator(from);

  for(int k = 0; k < 2*n; k++) {
   ensure(i.hasNext() == j.hasNext(), "Error (" + seed + "): divergence in hasNext() (iterator with starting point " +
from + ")");
   ensure(i.hasPrevious() == j.hasPrevious() , "Error (" + seed + "): divergence in hasPrevious() (iterator with starting
point " + from + ")");

   if (r.nextFloat() < .8 && i.hasNext()) {
    ensure(java.util.Arrays.equals((KEY_TYPE[])i.next(), (KEY_TYPE[])j.next()), "Error (" + seed + "): divergence in
next() (iterator with starting point " + from + ")");
    //System.err.println("Done next " + I + " " + J + "  " + badPrevious);

   }
   else if (r.nextFloat() < .2 && i.hasPrevious()) {
    ensure(java.util.Arrays.equals((KEY_TYPE[])i.previous(), (KEY_TYPE[])j.previous()), "Error (" + seed + "):
divergence in previous() (iterator with starting point " +
 from + ")");

   }
  }

 }



 try {
  java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
  java.io.OutputStream os = new java.io.FileOutputStream(ff);
  java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

  oos.writeObject(m);
  oos.close();

  java.io.InputStream is = new java.io.FileInputStream(ff);
  java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

  m = (ARRAY_FRONT_CODED_LIST)ois.readObject();
  ois.close();
  ff.delete();
 }
 catch(Exception e) {
  e.printStackTrace();
  System.exit(1);
 }
```

```
  ensure(contentEquals(m, t), "Error (" + seed + "): m does not equal t after save/read");

  System.out.println("Test OK");
  return;
 }


 public static void main(String args[]) {
  int n  = Integer.parseInt(args[1]);
  if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

  try {
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n,
 "speedComp".equals(args[0]));
   else if ("test".equals(args[0])) runTest(n);
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed: " + seed);
  }
 }

#endif


}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/ArrayFrontCodedList.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2010-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package PACKAGE;

import java.util.List;
import it.unimi.dsi.fastutil.BigList;
import it.unimi.dsi.fastutil.Size64;

#if ! KEY_CLASS_Reference

/** A type-specific {@link BigList}; provides some additional methods that use polymorphism to avoid (un)boxing.
*
* <p>Additionally, this interface strengthens {@link #iterator()},
 {@link #listIterator()},
* {@link #listIterator(long)} and {@link #subList(long,long)}.
*
* <p>Besides polymorphic methods, this interfaces specifies methods to copy into an array or remove contiguous
* sublists. Although the abstract implementation of this interface provides simple, one-by-one implementations
* of these methods, it is expected that concrete implementation override them with optimized versions.
*
* @see List
*/

public interface BIG_LIST KEY_GENERIC extends BigList<KEY_GENERIC_CLASS>, COLLECTION
KEY_GENERIC, Size64, Comparable<BigList<? extends KEY_GENERIC_CLASS>> {
#else

/** A type-specific {@link BigList}; provides some additional methods that use polymorphism to avoid (un)boxing.
*
* <p>Additionally, this interface strengthens {@link #listIterator()},
* {@link #listIterator(long)} and {@link #subList(long,long)}.
*
* <p>Besides polymorphic methods, this interfaces specifies methods to copy into an array or remove contiguous
* sublists. Although
 the abstract implementation of this interface provides simple, one-by-one implementations
* of these methods, it is expected that concrete implementation override them with optimized versions.
*
* @see List
*/

public interface BIG_LIST KEY_GENERIC extends BigList<KEY_GENERIC_CLASS>, COLLECTION
KEY_GENERIC, Size64 {
#endif

 /** Returns a type-specific iterator on the elements of this list.
  *
  * <p>Note that this specification strengthens the one given in {@link java.util.Collection#iterator()}.
  * @see java.util.Collection#iterator()
  */
```

```
@Override
KEY_BIG_LIST_ITERATOR KEY_GENERIC iterator();


/** Returns a type-specific big-list iterator on this type-specific big list.
 *
 * <p>Note that this specification strengthens the one given in {@link BigList#listIterator()}.
 * @see BigList#listIterator()
 */
@Override
KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator();

/** Returns a type-specific list iterator on this type-specific big list starting at a given index.

*
 * <p>Note that this specification strengthens the one given in {@link BigList#listIterator(long)}.
 * @see BigList#listIterator(long)
 */
@Override
KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(long index);

/** Returns a type-specific view of the portion of this type-specific big list from the index {@code from}, inclusive,
to the index {@code to}, exclusive.
 *
 * <p>Note that this specification strengthens the one given in {@link BigList#subList(long,long)}.
 *
 * @see BigList#subList(long,long)
 */
@Override
BIG_LIST KEY_GENERIC subList(long from, long to);

/** Copies (hopefully quickly) elements of this type-specific big list into the given big array.
 *
 * @param from the start index (inclusive).
 * @param a the destination big array.
 * @param offset the offset into the destination big array where to store the first element copied.
 * @param length the number of elements to be copied.
 */
void getElements(long from, KEY_TYPE a[][], long offset, long length);

/**
Removes (hopefully quickly) elements of this type-specific big list.
 *
 * @param from the start index (inclusive).
 * @param to the end index (exclusive).
 */
void removeElements(long from, long to);
```

```
 /** Add (hopefully quickly) elements to this type-specific big list.
  *
  * @param index the index at which to add elements.
  * @param a the big array containing the elements.
  */
 void addElements(long index, KEY_GENERIC_TYPE a[][]);

 /** Add (hopefully quickly) elements to this type-specific big list.
  *
  * @param index the index at which to add elements.
  * @param a the big array containing the elements.
  * @param offset the offset of the first element to add.
  * @param length the number of elements to add.
  */
 void addElements(long index, KEY_GENERIC_TYPE a[][], long offset, long length);

#if KEYS_PRIMITIVE

 /** Inserts the specified element at the specified position in this type-specific big list (optional operation).
  * @see BigList#add(long,Object)

 */
 void add(long index, KEY_TYPE key);

 /** Inserts all of the elements in the specified type-specific collection into this type-specific big list at the specified
position (optional operation).
  * @see List#addAll(int,java.util.Collection)
  */
 boolean addAll(long index, COLLECTION c);

 /** Inserts all of the elements in the specified type-specific big list into this type-specific big list at the specified
position (optional operation).
  * @see List#addAll(int,java.util.Collection)
  */
 boolean addAll(long index, BIG_LIST c);

 /** Appends all of the elements in the specified type-specific big list to the end of this type-specific big list (optional
operation).
  * @see List#addAll(int,java.util.Collection)
  */
 boolean addAll(BIG_LIST c);

 /** Returns the element at the specified position.
  * @see BigList#get(long)
  */
 KEY_TYPE GET_KEY(long index);

 /** Removes the element at the specified position.
```

```
 * @see BigList#remove(long) */
KEY_TYPE REMOVE_KEY(long index);

/**
Replaces the element at the specified position in this big list with the specified element (optional operation).
 * @see BigList#set(long,Object)
 */
KEY_TYPE set(long index, KEY_TYPE k);

/** Returns the index of the first occurrence of the specified element in this type-specific big list, or -1 if this big list
does not contain the element.
 * @see BigList#indexOf(Object)
 */
long indexOf(KEY_TYPE k);

/** Returns the index of the last occurrence of the specified element in this type-specific big list, or -1 if this big list
does not contain the element.
 * @see BigList#lastIndexOf(Object)
 */
long lastIndexOf(KEY_TYPE k);

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
void add(long index, KEY_CLASS key);

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
KEY_GENERIC_CLASS get(long index);

/** {@inheritDoc}
 * @deprecated
Please use the corresponding type-specific method instead. */
@Deprecated
@Override
long indexOf(Object o);

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
long lastIndexOf(Object o);

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
```

```
 @Deprecated
 @Override
 KEY_GENERIC_CLASS remove(long index);

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 KEY_GENERIC_CLASS set(long index, KEY_GENERIC_CLASS k);
#endif

}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/BigList.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import it.unimi.dsi.fastutil.objects.AbstractObjectSortedSet;
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;
import it.unimi.dsi.fastutil.objects.ObjectListIterator;
import it.unimi.dsi.fastutil.objects.ObjectSortedSet;

import VALUE_PACKAGE.VALUE_COLLECTION;
import VALUE_PACKAGE.VALUE_ABSTRACT_COLLECTION;
import
 VALUE_PACKAGE.VALUE_ITERATOR;

import java.util.Comparator;
```

```
import java.util.Iterator;
import java.util.Map;
import java.util.SortedMap;
import java.util.NoSuchElementException;

#if VALUES_PRIMITIVE
import VALUE_PACKAGE.VALUE_LIST_ITERATOR;
#endif

/** A type-specific red-black tree map with a fast, small-footprint implementation.
*
* <p>The iterators provided by the views of this class are type-specific {@linkplain
* it.unimi.dsi.fastutil.BidirectionalIterator bidirectional iterators}.
* Moreover, the iterator returned by {@code iterator()} can be safely cast
* to a type-specific {@linkplain java.util.ListIterator list iterator}.
*
*/

public class RB_TREE_MAP KEY_VALUE_GENERIC extends ABSTRACT_SORTED_MAP
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable {

 /** A reference to the root entry. */
 protected transient Entry KEY_VALUE_GENERIC tree;

 /** Number of entries in this map. */
 protected int count;

 /** The first key in this
 map. */
 protected transient Entry KEY_VALUE_GENERIC firstEntry;

 /** The last key in this map. */
 protected transient Entry KEY_VALUE_GENERIC lastEntry;

 /** Cached set of entries. */
 protected transient ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> entries;

 /** Cached set of keys. */
 protected transient SORTED_SET KEY_GENERIC keys;

 /** Cached collection of values. */
 protected transient VALUE_COLLECTION VALUE_GENERIC values;

 /** The value of this variable remembers, after a {@code put()}
  * or a {@code remove()}, whether the <em>domain</em> of the map
  * has been modified. */
 protected transient boolean modified;
```

```
/** This map's comparator, as provided in the constructor. */
protected Comparator<? super KEY_GENERIC_CLASS> storedComparator;

/** This map's actual comparator; it may differ from {@link #storedComparator} because it is
 always a type-specific comparator, so it could be derived from the former by wrapping. */
protected transient KEY_COMPARATOR KEY_SUPER_GENERIC
actualComparator;

private static final long serialVersionUID = -7046029254386353129L;

{
 allocatePaths();
}

/** Creates a new empty tree map.
 */

public RB_TREE_MAP() {
 tree = null;
 count = 0;
}

/** Generates the comparator that will be actually used.
 *
 * <p>When a given {@link Comparator} is specified and stored in {@link
 * #storedComparator}, we must check whether it is type-specific.  If it is
 * so, we can used directly, and we store it in {@link #actualComparator}. Otherwise,
 * we adapt it using a helper static method.
 */
private void setActualComparator() {
#if KEY_CLASS_Object
 actualComparator = storedComparator;
#else
 actualComparator = COMPARATORS.AS_KEY_COMPARATOR(storedComparator);
#endif
}


/** Creates a new empty tree map with the given comparator.
 *
 * @param c a (possibly type-specific) comparator.
 */

public RB_TREE_MAP(final Comparator<? super KEY_GENERIC_CLASS> c) {
 this();
 storedComparator = c;
 setActualComparator();
}
```

```
/**
Creates a new tree map copying a given map.
 *
 * @param m a {@link Map} to be copied into the new tree map.
 */

public RB_TREE_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS>
m) {
 this();
 putAll(m);
}

/** Creates a new tree map copying a given sorted map (and its {@link Comparator}).
 *
 * @param m a {@link SortedMap} to be copied into the new tree map.
 */

public RB_TREE_MAP(final SortedMap<KEY_GENERIC_CLASS,VALUE_GENERIC_CLASS> m) {
 this(m.comparator());
 putAll(m);
}

/** Creates a new tree map copying a given map.
 *
 * @param m a type-specific map to be copied into the new tree map.
 */

public RB_TREE_MAP(final MAP KEY_VALUE_EXTENDS_GENERIC m) {
 this();
 putAll(m);
}

/** Creates a new tree map copying a given sorted map (and its {@link Comparator}).
 *
 * @param m a type-specific sorted map to be copied into the new tree map.
 */

public RB_TREE_MAP(final SORTED_MAP KEY_VALUE_GENERIC m) {
 this(m.comparator());
 putAll(m);
}

/**
Creates a new tree map using the elements of two parallel arrays and the given comparator.
 *
 * @param k the array of keys of the new tree map.
```

```
 * @param v the array of corresponding values in the new tree map.
 * @param c a (possibly type-specific) comparator.
 * @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
 */

 public RB_TREE_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE v[], final
Comparator<? super KEY_GENERIC_CLASS> c) {
  this(c);
  if (k.length != v.length) throw new IllegalArgumentException("The key array and the value array have different
lengths (" + k.length + " and " + v.length + ")");
  for(int i = 0; i < k.length; i++) this.put(k[i], v[i]);
 }

 /** Creates a new tree map using the elements of two parallel arrays.
  *
  * @param k the array of keys of the new tree map.
  * @param v the array of corresponding values in the new tree map.
  * @throws IllegalArgumentException if {@code
 k} and {@code v} have different lengths.
  */

 public RB_TREE_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE v[]) {
  this(k, v, null);
 }

 /*
  * The following methods implements some basic building blocks used by
  * all accessors.  They are (and should be maintained) identical to those used in RBTreeSet.drv.
  *
  * The put()/remove() code is derived from Ben Pfaff's GNU libavl
  * (http://www.msu.edu/~pfaffben/avl/). If you want to understand what's
  * going on, you should have a look at the literate code contained therein
  * first.
  */



 /** Compares two keys in the right way.
  *
  * <p>This method uses the {@link #actualComparator} if it is non-{@code null}.
  * Otherwise, it resorts to primitive type comparisons or to {@link Comparable#compareTo(Object) compareTo()}.
  *
  * @param k1 the first key.
  * @param k2 the second key.
  * @return a number smaller than, equal to or greater than 0, as usual
  * (i.e., when k1 &lt; k2, k1 = k2 or k1 &gt; k2, respectively).

 */
```

```
SUPPRESS_WARNINGS_KEY_UNCHECKED
final int compare(final KEY_GENERIC_TYPE k1, final KEY_GENERIC_TYPE k2) {
 return actualComparator == null ? KEY_CMP(k1, k2) : actualComparator.compare(k1, k2);
}

/** Returns the entry corresponding to the given key, if it is in the tree; {@code null}, otherwise.
 *
 * @param k the key to search for.
 * @return the corresponding entry, or {@code null} if no entry with the given key exists.
 */

final Entry KEY_VALUE_GENERIC findKey(final KEY_GENERIC_TYPE k) {
 Entry KEY_VALUE_GENERIC e = tree;
 int cmp;

 while (e != null && (cmp = compare(k, e.key)) != 0) e = cmp < 0 ? e.left() : e.right();

 return e;
}

/** Locates a key.
 *
 * @param k a key.
 * @return the last entry on a search for the given key; this will be
 * the given key, if it present; otherwise, it will be either the smallest greater key or the greatest smaller key.
 */

final Entry KEY_VALUE_GENERIC locateKey(final KEY_GENERIC_TYPE k) {
 Entry KEY_VALUE_GENERIC
e = tree, last = tree;
 int cmp = 0;

 while (e != null && (cmp = compare(k, e.key)) != 0) {
  last = e;
  e = cmp < 0 ? e.left() : e.right();
 }

 return cmp == 0 ? e : last;
}

/** This vector remembers the path and the direction followed during the
 *  current insertion. It suffices for about 2<sup>32</sup> entries. */
private transient boolean dirPath[];
private transient Entry KEY_VALUE_GENERIC nodePath[];

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED_RAWTYPES
private void allocatePaths() {
 dirPath = new boolean[64];
```

```
 nodePath = new Entry[64];
 }

#if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
 /** Adds an increment to value currently associated with a key.
  *
  * <p>Note that this method respects the {@linkplain #defaultReturnValue() default return value} semantics: when
  * called with a key that does not currently appears in the map, the key
  * will be associated with the default return value plus
  * the given increment.
  *
  * @param k the key.
  * @param incr
 the increment.
  * @return the old value, or the {@linkplain #defaultReturnValue() default return value} if no value was present for
the given key.
  */
 public VALUE_GENERIC_TYPE addTo(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE incr) {
  Entry KEY_VALUE_GENERIC e = add(k);
  final VALUE_GENERIC_TYPE oldValue = e.value;
  e.value += incr;
  return oldValue;
 }
#endif

 @Override
 public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
  Entry KEY_VALUE_GENERIC e = add(k);
  final VALUE_GENERIC_TYPE oldValue = e.value;
  e.value = v;
  return oldValue;
 }

 /** Returns a node with key k in the balanced tree, creating one with defRetValue if necessary.
  *
  * @param k the key
  * @return a node with key k. If a node with key k already exists, then that node is returned,
  * otherwise a new node with defRetValue is created ensuring that the tree is balanced
  * after creation of the node.
  */
 private Entry KEY_VALUE_GENERIC add(final KEY_GENERIC_TYPE
 k) {
  /* After execution of this method, modified is true iff a new entry has been inserted. */
  modified = false;
  int maxDepth = 0;

  Entry KEY_VALUE_GENERIC e;
```

```
 if (tree == null) { // The case of the empty tree is treated separately.
  count++;
  e = tree = lastEntry = firstEntry = new Entry KEY_VALUE_GENERIC_DIAMOND(k, defRetValue);
 }
 else {
  Entry KEY_VALUE_GENERIC p = tree;
  int cmp, i = 0;

  while(true) {
   if ((cmp = compare(k, p.key)) == 0) {
    // We clean up the node path, or we could have stale references later.
    while(i-- != 0) nodePath[i] = null;
    return p;
   }

   nodePath[i] = p;

   if (dirPath[i++] = cmp > 0) {
    if (p.succ()) {
     count++;
     e = new Entry KEY_VALUE_GENERIC_DIAMOND(k, defRetValue);

     if (p.right == null) lastEntry = e;

     e.left = p;
     e.right = p.right;

     p.right(e);

     break;
    }

    p = p.right;
   }
   else {
    if (p.pred()) {
     count++;
     e = new Entry KEY_VALUE_GENERIC_DIAMOND(k,
defRetValue);

     if (p.left == null) firstEntry = e;

     e.right = p;
     e.left = p.left;

     p.left(e);

     break;
```

```
    }

    p = p.left;
   }
  }

  modified = true;
  maxDepth = i--;

  while(i > 0 && ! nodePath[i].black()) {
   if (! dirPath[i - 1]) {
    Entry KEY_VALUE_GENERIC y = nodePath[i - 1].right;

    if (! nodePath[i - 1].succ() && ! y.black()) {
     nodePath[i].black(true);
     y.black(true);
     nodePath[i - 1].black(false);
     i -= 2;
    }
    else {
     Entry KEY_VALUE_GENERIC x;

     if (! dirPath[i]) y = nodePath[i];
     else {
      x = nodePath[i];
      y = x.right;
      x.right = y.left;
      y.left = x;
      nodePath[i - 1].left = y;

      if (y.pred()) {
       y.pred(false);
       x.succ(y);
      }
     }

     x = nodePath[i - 1];
     x.black(false);
     y.black(true);

     x.left = y.right;
     y.right = x;
     if (i < 2) tree = y;
     else {
      if (dirPath[i
- 2]) nodePath[i - 2].right = y;
       else nodePath[i - 2].left = y;
     }
```

```
  if (y.succ()) {
   y.succ(false);
   x.pred(y);
  }
  break;
 }
}
else {
 Entry KEY_VALUE_GENERIC y = nodePath[i - 1].left;

 if (! nodePath[i - 1].pred() && ! y.black()) {
  nodePath[i].black(true);
  y.black(true);
  nodePath[i - 1].black(false);
  i -= 2;
 }
 else {
  Entry KEY_VALUE_GENERIC x;

  if (dirPath[i]) y = nodePath[i];
  else {
   x = nodePath[i];
   y = x.left;
   x.left = y.right;
   y.right = x;
   nodePath[i - 1].right = y;

   if (y.succ()) {
    y.succ(false);
    x.pred(y);
   }

  }

  x = nodePath[i - 1];
  x.black(false);
  y.black(true);

  x.right = y.left;
  y.left = x;
  if (i < 2) tree = y;
  else {
   if (dirPath[i - 2]) nodePath[i - 2].right = y;
   else nodePath[i - 2].left = y;
  }

  if (y.pred()){
```

```
      y.pred(false);
      x.succ(y);
     }

     break;
    }
   }
  }
 }
 tree.black(true);
 //
We clean up the node path, or we could have stale references later.
 while(maxDepth-- != 0) nodePath[maxDepth] = null;
 return e;
}


/* After execution of this method, {@link #modified} is true iff an entry
has been deleted. */

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) {
 modified = false;

 if (tree == null) return defRetValue;

 Entry KEY_VALUE_GENERIC p = tree;
 int cmp;
 int i = 0;
 final KEY_GENERIC_TYPE kk = KEY_GENERIC_CAST k;

 while(true) {
  if ((cmp = compare(kk, p.key)) == 0) break;

  dirPath[i] = cmp > 0;
  nodePath[i] = p;

  if (dirPath[i++]) {
   if ((p = p.right()) == null) {
    // We clean up the node path, or we could have stale references later.
    while(i-- != 0) nodePath[i] = null;
    return defRetValue;
   }
  }
  else {
   if ((p = p.left()) == null) {
    // We clean up the node
```

```
path, or we could have stale references later.
    while(i-- != 0) nodePath[i] = null;
    return defRetValue;
   }
  }

 }

 if (p.left == null) firstEntry = p.next();
 if (p.right == null) lastEntry = p.prev();

 if (p.succ()) {
  if (p.pred()) {
   if (i == 0) tree = p.left;
   else {
    if (dirPath[i - 1]) nodePath[i - 1].succ(p.right);
    else nodePath[i - 1].pred(p.left);
   }
  }
  else {
   p.prev().right = p.right;

   if (i == 0) tree = p.left;
   else {
    if (dirPath[i - 1]) nodePath[i - 1].right = p.left;
    else nodePath[i - 1].left = p.left;
   }
  }
 }
 else {
  boolean color;
  Entry KEY_VALUE_GENERIC r = p.right;

  if (r.pred()) {
   r.left = p.left;
   r.pred(p.pred());
   if (! r.pred()) r.prev().right = r;
   if (i == 0) tree = r;
   else {
    if (dirPath[i - 1]) nodePath[i - 1].right = r;
    else nodePath[i - 1].left = r;
   }

   color = r.black();
   r.black(p.black());
   p.black(color);
   dirPath[i] = true;
   nodePath[i++]
```

```
= r;
  }
  else {
   Entry KEY_VALUE_GENERIC s;
   int j = i++;

   while(true) {
    dirPath[i] = false;
    nodePath[i++] = r;
    s = r.left;
    if (s.pred()) break;
    r = s;
   }

   dirPath[j] = true;
   nodePath[j] = s;

   if (s.succ()) r.pred(s);
   else r.left = s.right;

   s.left = p.left;

   if (! p.pred()) {
    p.prev().right = s;
    s.pred(false);
   }

   s.right(p.right);

   color = s.black();
   s.black(p.black());
   p.black(color);

   if (j == 0) tree = s;
   else {
    if (dirPath[j - 1]) nodePath[j - 1].right = s;
    else nodePath[j - 1].left = s;
   }
  }
 }

 int maxDepth = i;

 if (p.black()) {
  for(; i > 0; i--) {
   if (dirPath[i - 1] && ! nodePath[i - 1].succ() ||
     ! dirPath[i - 1] && ! nodePath[i - 1].pred()) {
    Entry KEY_VALUE_GENERIC x = dirPath[i - 1] ? nodePath[i - 1].right : nodePath[i - 1].left;
```

```
    if (! x.black()) {
     x.black(true);
     break;
    }
   }

   if (! dirPath[i
- 1]) {
    Entry KEY_VALUE_GENERIC w = nodePath[i - 1].right;

    if (! w.black()) {
     w.black(true);
     nodePath[i - 1].black(false);

     nodePath[i - 1].right = w.left;
     w.left = nodePath[i - 1];

     if (i < 2) tree = w;
     else {
      if (dirPath[i - 2]) nodePath[i - 2].right = w;
      else nodePath[i - 2].left = w;
     }

     nodePath[i] = nodePath[i - 1];
     dirPath[i] = false;
     nodePath[i - 1] = w;
     if (maxDepth == i++) maxDepth++;

     w = nodePath[i - 1].right;
    }

    if ((w.pred() || w.left.black()) &&
      (w.succ() || w.right.black())) {
     w.black(false);
    }
    else {
     if (w.succ() || w.right.black()) {
      Entry KEY_VALUE_GENERIC y = w.left;

      y.black (true);
      w.black(false);
      w.left = y.right;
      y.right = w;
      w = nodePath[i - 1].right = y;

      if (w.succ()) {
       w.succ(false);
```

```
      w.right.pred(w);
     }
    }

    w.black(nodePath[i - 1].black());
    nodePath[i
- 1].black(true);
    w.right.black(true);

    nodePath[i - 1].right = w.left;
    w.left = nodePath[i - 1];

    if (i < 2) tree = w;
    else {
     if (dirPath[i - 2]) nodePath[i - 2].right = w;
     else nodePath[i - 2].left = w;
    }

    if (w.pred()) {
     w.pred(false);
     nodePath[i - 1].succ(w);
    }
    break;
   }
  }
  else {
   Entry KEY_VALUE_GENERIC w = nodePath[i - 1].left;

   if (! w.black()) {
    w.black (true);
    nodePath[i - 1].black(false);

    nodePath[i - 1].left = w.right;
    w.right = nodePath[i - 1];

    if (i < 2) tree = w;
    else {
     if (dirPath[i - 2]) nodePath[i - 2].right = w;
     else nodePath[i - 2].left = w;
    }

    nodePath[i] = nodePath[i - 1];
    dirPath[i] = true;
    nodePath[i - 1] = w;
    if (maxDepth == i++) maxDepth++;

    w = nodePath[i - 1].left;
   }
```

```
    if ((w.pred() || w.left.black()) &&
      (w.succ() || w.right.black())) {
     w.black(false);
    }
    else
{
     if (w.pred() || w.left.black()) {
      Entry KEY_VALUE_GENERIC y = w.right;

      y.black(true);
      w.black (false);
      w.right = y.left;
      y.left = w;
      w = nodePath[i - 1].left = y;

      if (w.pred()) {
       w.pred(false);
       w.left.succ(w);
      }
     }

     w.black(nodePath[i - 1].black());
     nodePath[i - 1].black(true);
     w.left.black(true);

     nodePath[i - 1].left = w.right;
     w.right = nodePath[i - 1];

     if (i < 2) tree = w;
     else {
      if (dirPath[i - 2]) nodePath[i - 2].right = w;
      else nodePath[i - 2].left = w;
     }

     if (w.succ()) {
      w.succ(false);
      nodePath[i - 1].pred(w);
     }
     break;
    }
   }
  }

 if (tree != null) tree.black(true);
}

modified = true;
```

```
 count--;
 // We clean up the node path, or we could have stale references later.
 while(maxDepth-- != 0) nodePath[maxDepth] = null;
 return p.value;
}


@Override
public
boolean containsValue(final VALUE_TYPE v) {
 final ValueIterator i = new ValueIterator();
 VALUE_TYPE ev;

 int j = count;
 while(j-- != 0) {
  ev = i.NEXT_VALUE();
  if (VALUE_EQUALS(ev, v)) return true;
 }

 return false;
}


@Override
public void clear() {
 count = 0;
 tree = null;
 entries = null;
 values = null;
 keys = null;
 firstEntry = lastEntry = null;
}


/** This class represent an entry in a tree map.
 *
 * <p>We use the only "metadata", i.e., {@link Entry#info}, to store
 * information about color, predecessor status and successor status.
 *
 * <p>Note that since the class is recursive, it can be
 * considered equivalently a tree.
 */

 private static final class Entry KEY_VALUE_GENERIC extends ABSTRACT_MAP.BasicEntry
KEY_VALUE_GENERIC implements Cloneable {
 /** The the bit in this mask is true, the node is black. */
 private static final int BLACK_MASK = 1;
 /** If the bit in this mask is true, {@link #right} points to a successor.
```

```
 */
  private static final int SUCC_MASK = 1 << 31;
  /** If the bit in this mask is true, {@link #left} points to a predecessor. */
  private static final int PRED_MASK = 1 << 30;
  /** The pointers to the left and right subtrees. */
  Entry KEY_VALUE_GENERIC left, right;
  /** This integers holds different information in different bits (see {@link #SUCC_MASK} and {@link
#PRED_MASK}. */
  int info;

  Entry() {
   super(KEY_NULL, VALUE_NULL);
  }

  /** Creates a new entry with the given key and value.
   *
   * @param k a key.
   * @param v a value.
   */
  Entry(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
   super(k, v);
   info = SUCC_MASK | PRED_MASK;
  }

  /** Returns the left subtree.
   *
   * @return the left subtree ({@code null} if the left
   * subtree is empty).
   */
  Entry KEY_VALUE_GENERIC left() {
   return (info & PRED_MASK) != 0 ? null : left;
  }

  /** Returns the right subtree.
   *
   * @return the right subtree ({@code null} if the right
   * subtree
 is empty).
   */
  Entry KEY_VALUE_GENERIC right() {
   return (info & SUCC_MASK) != 0 ? null : right;
  }

  /** Checks whether the left pointer is really a predecessor.
   * @return true if the left pointer is a predecessor.
   */
  boolean pred() {
   return (info & PRED_MASK) != 0;
```

```
}

/** Checks whether the right pointer is really a successor.
 * @return true if the right pointer is a successor.
 */
boolean succ() {
 return (info & SUCC_MASK) != 0;
}

/** Sets whether the left pointer is really a predecessor.
 * @param pred if true then the left pointer will be considered a predecessor.
 */
void pred(final boolean pred) {
 if (pred) info |= PRED_MASK;
 else info &= ~PRED_MASK;
}

/** Sets whether the right pointer is really a successor.
 * @param succ if true then the right pointer will be considered a successor.
 */
void succ(final boolean succ) {
 if (succ) info |= SUCC_MASK;
 else info &= ~SUCC_MASK;
}

/** Sets the left pointer to a
predecessor.
 * @param pred the predecessr.
 */
void pred(final Entry KEY_VALUE_GENERIC pred) {
 info |= PRED_MASK;
 left = pred;
}

/** Sets the right pointer to a successor.
 * @param succ the successor.
 */
void succ(final Entry KEY_VALUE_GENERIC succ) {
 info |= SUCC_MASK;
 right = succ;
}

/** Sets the left pointer to the given subtree.
 * @param left the new left subtree.
 */
void left(final Entry KEY_VALUE_GENERIC left) {
 info &= ~PRED_MASK;
 this.left = left;
```

```
}

/** Sets the right pointer to the given subtree.
 * @param right the new right subtree.
 */
void right(final Entry KEY_VALUE_GENERIC right) {
 info &= ~SUCC_MASK;
 this.right = right;
}


/** Returns whether this node is black.
 * @return true iff this node is black.
 */
boolean black() {
 return (info & BLACK_MASK) != 0;
}

/** Sets whether this node is black.
 * @param black if true, then this node becomes black; otherwise, it becomes red..
 */
void
black(final boolean black) {
 if (black) info |= BLACK_MASK;
 else info &= ~BLACK_MASK;
}

/** Computes the next entry in the set order.
 *
 * @return the next entry ({@code null}) if this is the last entry).
 */

Entry KEY_VALUE_GENERIC next() {
 Entry KEY_VALUE_GENERIC next = this.right;
 if ((info & SUCC_MASK) == 0) while ((next.info & PRED_MASK) == 0) next = next.left;
 return next;
}

/** Computes the previous entry in the set order.
 *
 * @return the previous entry ({@code null}) if this is the first entry).
 */

Entry KEY_VALUE_GENERIC prev() {
 Entry KEY_VALUE_GENERIC prev = this.left;
 if ((info & PRED_MASK) == 0) while ((prev.info & SUCC_MASK) == 0) prev = prev.right;
 return prev;
}
```

```
 @Override
 public VALUE_GENERIC_TYPE setValue(final VALUE_GENERIC_TYPE value) {
  final VALUE_GENERIC_TYPE oldValue = this.value;
  this.value = value;
  return oldValue;
 }

 @Override
 SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
 public Entry
KEY_VALUE_GENERIC clone() {
  Entry KEY_VALUE_GENERIC c;
  try {
   c = (Entry KEY_VALUE_GENERIC)super.clone();
  }
  catch(CloneNotSupportedException cantHappen) {
   throw new InternalError();
  }

  c.key = key;
  c.value = value;
  c.info = info;

  return c;
 }

 @Override
 @SuppressWarnings("unchecked")
 public boolean equals(final Object o) {
  if (!(o instanceof Map.Entry)) return false;
  Map.Entry <KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS> e = (Map.Entry <KEY_GENERIC_CLASS,
VALUE_GENERIC_CLASS>)o;

  return KEY_EQUALS(key, KEY_CLASS2TYPE(e.getKey())) && VALUE_EQUALS(value,
VALUE_CLASS2TYPE(e.getValue()));
 }

 @Override
 public int hashCode() {
  return KEY2JAVAHASH_NOT_NULL(key) ^ VALUE2JAVAHASH(value);
 }


 @Override
 public String toString() {
  return key + "=>" + value;
 }
```

```
 /*
   public void prettyPrint() {
   prettyPrint(0);
   }

   public void prettyPrint(int level) {
   if (pred()) {
   for (int i = 0; i < level; i++)
   System.err.print("
 ");
   System.err.println("pred: " + left);
   }
   else if (left != null)
   left.prettyPrint(level +1);
   for (int i = 0; i < level; i++)
   System.err.print("  ");
   System.err.println(key + "=" + value + " (" + balance() + ")");
   if (succ()) {
   for (int i = 0; i < level; i++)
   System.err.print("  ");
   System.err.println("succ: " + right);
   }
   else if (right != null)
   right.prettyPrint(level + 1);
   }*/
}

/*
  public void prettyPrint() {
  System.err.println("size: " + count);
  if (tree != null) tree.prettyPrint();
  }*/

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public boolean containsKey(final KEY_TYPE k) {
 return findKey(KEY_GENERIC_CAST k) != null;
}

@Override
public int size() {
 return count;
}

@Override
public boolean isEmpty() {
 return count == 0;
```

```
 }

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 @Override
 public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
  final Entry KEY_VALUE_GENERIC e = findKey(KEY_GENERIC_CAST
 k);
  return e == null ? defRetValue : e.value;
 }

 @Override
 public KEY_GENERIC_TYPE FIRST_KEY() {
  if (tree == null) throw new NoSuchElementException();
  return firstEntry.key;
 }

 @Override
 public KEY_GENERIC_TYPE LAST_KEY() {
  if (tree == null) throw new NoSuchElementException();
  return lastEntry.key;
 }


 /** An abstract iterator on the whole range.
  *
  * <p>This class can iterate in both directions on a threaded tree.
  */

 private class TreeIterator {
  /** The entry that will be returned by the next call to {@link java.util.ListIterator#previous()} (or {@code null} if
no previous entry exists). */
  Entry KEY_VALUE_GENERIC prev;
  /** The entry that will be returned by the next call to {@link java.util.ListIterator#next()} (or {@code null} if no
next entry exists). */
  Entry KEY_VALUE_GENERIC next;
  /** The last entry that was returned (or {@code null} if we did not iterate or used {@link #remove()}). */
  Entry KEY_VALUE_GENERIC curr;
  /** The current index (in
 the sense of a {@link java.util.ListIterator}). Note that this value is not meaningful when this {@link TreeIterator}
has been created using the nonempty constructor.*/
  int index = 0;

  TreeIterator() {
   next = firstEntry;
  }

  TreeIterator(final KEY_GENERIC_TYPE k) {
   if ((next = locateKey(k)) != null) {
```

```
  if (compare(next.key, k) <= 0) {
   prev = next;
   next = next.next();
  }
  else prev = next.prev();
 }
}

public boolean hasNext() { return next != null; }
public boolean hasPrevious() { return prev != null; }

void updateNext() {
 next = next.next();
}

Entry KEY_VALUE_GENERIC nextEntry() {
 if (! hasNext()) throw new NoSuchElementException();
 curr = prev = next;
 index++;
 updateNext();
 return curr;
}

void updatePrevious() {
 prev = prev.prev();
}

Entry KEY_VALUE_GENERIC previousEntry() {
 if (! hasPrevious()) throw new NoSuchElementException();
 curr = next = prev;
 index--;
 updatePrevious();
 return curr;
}

public
int nextIndex() {
 return index;
}

public int previousIndex() {
 return index - 1;
}

public void remove() {
 if (curr == null) throw new IllegalStateException();
 /* If the last operation was a next(), we are removing an entry that preceeds
    the current index, and thus we must decrement it. */
```

```
  if (curr == prev) index--;
  next = prev = curr;
  updatePrevious();
  updateNext();
  RB_TREE_MAP.this.REMOVE_VALUE(curr.key);
  curr = null;
 }

 public int skip(final int n) {
  int i = n;
  while(i-- != 0 && hasNext()) nextEntry();
  return n - i - 1;
 }

 public int back(final int n) {
  int i = n;
  while(i-- != 0 && hasPrevious()) previousEntry();
  return n - i - 1;
 }
}


/** An iterator on the whole range.
 *
 * <p>This class can iterate in both directions on a threaded tree.
 */

 private class EntryIterator extends TreeIterator implements ObjectListIterator<MAP.Entry
KEY_VALUE_GENERIC> {
  EntryIterator() {}

  EntryIterator(final
 KEY_GENERIC_TYPE k) {
   super(k);
  }

  @Override
  public MAP.Entry KEY_VALUE_GENERIC next() { return nextEntry(); }
  @Override
  public MAP.Entry KEY_VALUE_GENERIC previous() { return previousEntry(); }
 }


 @Override
 public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() {
  if (entries == null) entries = new AbstractObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC>() {
    final Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator =  (Comparator<MAP.Entry
KEY_VALUE_GENERIC>) (x, y) -> RB_TREE_MAP.this.actualComparator.compare(x.ENTRY_GET_KEY(),
```

```
y.ENTRY_GET_KEY());

    @Override
    public Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator() { return comparator; }

    @Override
    public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() { return new EntryIterator(); }

    @Override
    public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator(final MAP.Entry
KEY_VALUE_GENERIC from) { return new EntryIterator(from.ENTRY_GET_KEY());
 }

    @Override
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public boolean contains(final Object o) {
     if (!(o instanceof Map.Entry)) return false;
     final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
     if (e.getKey() == null || ! (e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
     if (e.getValue() == null || ! (e.getValue() instanceof VALUE_CLASS)) return false;
#endif
     final Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey()));
     return e.equals(f);
    }

    @Override
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public boolean remove(final Object o) {
     if (!(o instanceof Map.Entry)) return false;
     final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
     if (e.getKey() == null || ! (e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
     if (e.getValue() == null || ! (e.getValue() instanceof VALUE_CLASS)) return false;
#endif
     final Entry KEY_VALUE_GENERIC
 f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey()));
     if (f == null || ! VALUE_EQUALS(f.ENTRY_GET_VALUE(), VALUE_OBJ2TYPE(e.getValue()))) return false;
     RB_TREE_MAP.this.REMOVE_VALUE(f.key);
     return true;
    }

    @Override
    public int size() { return count; }
```

```
    @Override
    public void clear() { RB_TREE_MAP.this.clear(); }

    @Override
    public MAP.Entry KEY_VALUE_GENERIC first() { return firstEntry; }

    @Override
    public MAP.Entry KEY_VALUE_GENERIC last() { return lastEntry; }

    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> subSet(MAP.Entry KEY_VALUE_GENERIC
from, MAP.Entry KEY_VALUE_GENERIC to) { return subMap(from.ENTRY_GET_KEY(),
to.ENTRY_GET_KEY()).ENTRYSET(); }

    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> headSet(MAP.Entry KEY_VALUE_GENERIC
to) { return headMap(to.ENTRY_GET_KEY()).ENTRYSET(); }

    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> tailSet(MAP.Entry KEY_VALUE_GENERIC
 from) { return tailMap(from.ENTRY_GET_KEY()).ENTRYSET(); }
   };

  return entries;
  }

  /** An iterator on the whole range of keys.
   *
   * <p>This class can iterate in both directions on the keys of a threaded tree. We
   * simply override the {@link java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()} methods (and
possibly
   * their type-specific counterparts) so that they return keys instead of entries.
   */
  private final class KeyIterator extends TreeIterator implements KEY_LIST_ITERATOR KEY_GENERIC {
   public KeyIterator() {}
   public KeyIterator(final KEY_GENERIC_TYPE k) { super(k); }

   @Override
   public KEY_GENERIC_TYPE NEXT_KEY() { return nextEntry().key; }

   @Override
   public KEY_GENERIC_TYPE PREV_KEY() { return previousEntry().key; }
  };


  /** A keyset implementation using a more direct implementation for iterators. */
  private class KeySet extends ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC.KeySet {
```

```
 @Override
 public KEY_BIDI_ITERATOR KEY_GENERIC
iterator() { return new KeyIterator(); }
 @Override
 public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
KeyIterator(from); }
}

/** Returns a type-specific sorted set view of the keys contained in this map.
 *
 * <p>In addition to the semantics of {@link java.util.Map#keySet()}, you can
 * safely cast the set returned by this call to a type-specific sorted
 * set interface.
 *
 * @return a type-specific sorted set view of the keys contained in this map.
 */
@Override
public SORTED_SET KEY_GENERIC keySet() {
 if (keys == null) keys = new KeySet();
 return keys;
}

/** An iterator on the whole range of values.
 *
 * <p>This class can iterate in both directions on the values of a threaded tree. We
 * simply override the {@link java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()} methods (and
possibly
 * their type-specific counterparts) so that they return values instead of entries.
 */
private final
class ValueIterator extends TreeIterator implements VALUE_LIST_ITERATOR VALUE_GENERIC {
 @Override
 public VALUE_GENERIC_TYPE NEXT_VALUE() { return nextEntry().value; }

 @Override
 public VALUE_GENERIC_TYPE PREV_VALUE() { return previousEntry().value; }
};

/** Returns a type-specific collection view of the values contained in this map.
 *
 * <p>In addition to the semantics of {@link java.util.Map#values()}, you can
 * safely cast the collection returned by this call to a type-specific collection
 * interface.
 *
 * @return a type-specific collection view of the values contained in this map.
 */
@Override
public VALUE_COLLECTION VALUE_GENERIC values() {
```

```
 if (values == null) values = new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {
   @Override
   public VALUE_ITERATOR VALUE_GENERIC iterator() { return new ValueIterator(); }
   @Override
   public boolean contains(final VALUE_TYPE k) { return containsValue(k); }
   @Override
   public int size() { return count; }
   @Override
   public
 void clear() { RB_TREE_MAP.this.clear(); }
  };

 return values;
 }

 @Override
 public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return actualComparator; }

 @Override
 public SORTED_MAP KEY_VALUE_GENERIC headMap(KEY_GENERIC_TYPE to) { return new
Submap(KEY_NULL, true, to, false); }

 @Override
 public SORTED_MAP KEY_VALUE_GENERIC tailMap(KEY_GENERIC_TYPE from) { return new
Submap(from, false, KEY_NULL, true); }

 @Override
 public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE
to) { return new Submap(from, false, to, false); }

 /** A submap with given range.
  *
  * <p>This class represents a submap. One has to specify the left/right
  * limits (which can be set to -&infin; or &infin;). Since the submap is a
  * view on the map, at a given moment it could happen that the limits of
  * the range are not any longer in the main map. Thus, things such as
  * {@link java.util.SortedMap#firstKey()} or {@link java.util.Collection#size()}
 must be always computed
  * on-the-fly.
  */
 private final class Submap extends ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC implements
java.io.Serializable {
  private static final long serialVersionUID = -7046029254386353129L;

  /** The start of the submap range, unless {@link #bottom} is true. */
  KEY_GENERIC_TYPE from;
  /** The end of the submap range, unless {@link #top} is true. */
  KEY_GENERIC_TYPE to;
```

```
 /** If true, the submap range starts from -&infin;. */
 boolean bottom;
 /** If true, the submap range goes to &infin;. */
 boolean top;
 /** Cached set of entries. */
 protected transient ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> entries;
 /** Cached set of keys. */
 protected transient SORTED_SET KEY_GENERIC keys;
 /** Cached collection of values. */
 protected transient VALUE_COLLECTION VALUE_GENERIC values;

 /** Creates a new submap with given key range.
  *
  * @param from the start of the submap range.
  * @param bottom if true, the first parameter
is ignored and the range starts from -&infin;.
  * @param to the end of the submap range.
  * @param top if true, the third parameter is ignored and the range goes to &infin;.
  */
 public Submap(final KEY_GENERIC_TYPE from, final boolean bottom, final KEY_GENERIC_TYPE to, final
boolean top) {
  if (! bottom && ! top && RB_TREE_MAP.this.compare(from, to) > 0) throw new
IllegalArgumentException("Start key (" + from  + ") is larger than end key (" + to + ")");

  this.from = from;
  this.bottom = bottom;
  this.to = to;
  this.top = top;
  this.defRetValue = RB_TREE_MAP.this.defRetValue;
 }

 @Override
 public void clear() {
  final SubmapIterator i = new SubmapIterator();
  while(i.hasNext()) {
   i.nextEntry();
   i.remove();
  }
 }

 /** Checks whether a key is in the submap range.
  * @param k a key.
  * @return true if is the key is in the submap range.
  */
 final boolean in(final KEY_GENERIC_TYPE k) {
  return (bottom || RB_TREE_MAP.this.compare(k, from)
>= 0) &&
   (top || RB_TREE_MAP.this.compare(k, to) < 0);
```

```
  }

  @Override
  public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() {
   if (entries == null) entries = new AbstractObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC>() {
    @Override
    public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() {
     return new SubmapEntryIterator();
    }

    @Override
    public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator(final MAP.Entry
KEY_VALUE_GENERIC from) {
     return new SubmapEntryIterator(from.ENTRY_GET_KEY());
    }

    @Override
    public Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator() { return
RB_TREE_MAP.this.ENTRYSET().comparator(); }

    @Override
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public boolean contains(final Object o) {
     if (!(o instanceof Map.Entry)) return false;
     final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
     if (e.getKey() == null || ! (e.getKey() instanceof
 KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
     if (e.getValue() == null || ! (e.getValue() instanceof VALUE_CLASS)) return false;
#endif
     final RB_TREE_MAP.Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST
e.getKey()));
     return f != null && in(f.key) && e.equals(f);
    }

    @Override
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public boolean remove(final Object o) {
     if (!(o instanceof Map.Entry)) return false;
     final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
     if (e.getKey() == null || ! (e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
     if (e.getValue() == null || ! (e.getValue() instanceof VALUE_CLASS)) return false;
#endif
```

```
      final RB_TREE_MAP.Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST
e.getKey()));
      if (f != null && in(f.key)) Submap.this.REMOVE_VALUE(f.key);
      return f != null;
     }

     @Override
     public int size() {
      int c =
 0;
      for(Iterator<?> i = iterator(); i.hasNext(); i.next()) c++;
      return c;
     }

     @Override
     public boolean isEmpty() { return ! new SubmapIterator().hasNext(); }

     @Override
     public void clear() { Submap.this.clear(); }

     @Override
     public MAP.Entry KEY_VALUE_GENERIC first() { return firstEntry(); }

     @Override
     public MAP.Entry KEY_VALUE_GENERIC last() { return lastEntry(); }

     @Override
     public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> subSet(MAP.Entry KEY_VALUE_GENERIC
from, MAP.Entry KEY_VALUE_GENERIC to) { return subMap(from.ENTRY_GET_KEY(),
to.ENTRY_GET_KEY()).ENTRYSET(); }

     @Override
     public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> headSet(MAP.Entry KEY_VALUE_GENERIC
to) { return headMap(to.ENTRY_GET_KEY()).ENTRYSET(); }

     @Override
     public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> tailSet(MAP.Entry KEY_VALUE_GENERIC
from) { return tailMap(from.ENTRY_GET_KEY()).ENTRYSET(); }
    };

   return entries;
  }

  private
 class KeySet extends ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC.KeySet {
   @Override
   public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new SubmapKeyIterator(); }
   @Override
```

```
  public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
SubmapKeyIterator(from); }
 }

 @Override
 public SORTED_SET KEY_GENERIC keySet() {
  if (keys == null) keys = new KeySet();
  return keys;
 }

 @Override
 public VALUE_COLLECTION VALUE_GENERIC values() {
  if (values == null) values = new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {
   @Override
   public VALUE_ITERATOR VALUE_GENERIC iterator() { return new SubmapValueIterator(); }
   @Override
   public boolean contains(final VALUE_TYPE k) { return containsValue(k); }
   @Override
   public int size() { return Submap.this.size(); }
   @Override
   public void clear() { Submap.this.clear(); }
  };

  return values;
 }

   @Override
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public
 boolean containsKey(final KEY_TYPE k) { return in(KEY_GENERIC_CAST k) &&
RB_TREE_MAP.this.containsKey(k); }

 @Override
 public boolean containsValue(final VALUE_TYPE v) {
  final SubmapIterator i = new SubmapIterator();
  VALUE_TYPE ev;

  while(i.hasNext()) {
   ev = i.nextEntry().value;
   if (VALUE_EQUALS(ev, v)) return true;
  }

  return false;
 }

 @Override
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
```

```
  final RB_TREE_MAP.Entry KEY_VALUE_GENERIC e;
  final KEY_GENERIC_TYPE kk = KEY_GENERIC_CAST k;
  return in(kk) && (e = findKey(kk)) != null ? e.value : this.defRetValue;
 }

 @Override
 public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
  modified = false;
  if (! in(k)) throw new IllegalArgumentException("Key (" + k + ") out of range [" + (bottom ? "-" :
String.valueOf(from)) + ", " + (top ? "-" : String.valueOf(to)) + ")");
  final VALUE_GENERIC_TYPE oldValue = RB_TREE_MAP.this.put(k,
 v);
  return modified ? this.defRetValue : oldValue;
 }

 @Override
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) {
  modified = false;
  if (! in(KEY_GENERIC_CAST k)) return this.defRetValue;
  final VALUE_GENERIC_TYPE oldValue = RB_TREE_MAP.this.REMOVE_VALUE(k);
  return modified ? oldValue : this.defRetValue;
 }

 @Override
 public int size() {
  final SubmapIterator i = new SubmapIterator();
  int n = 0;

  while(i.hasNext()) {
   n++;
   i.nextEntry();
  }

  return n;
 }

 @Override
 public boolean isEmpty() { return ! new SubmapIterator().hasNext(); }

 @Override
 public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return actualComparator; }

 @Override
 public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_TYPE to) {
  if (top) return new Submap(from, bottom, to, false);
  return compare(to, this.to) < 0 ? new Submap(from, bottom, to, false) : this;
 }
```

```
 @Override
 public SORTED_MAP
KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_TYPE from) {
  if (bottom) return new Submap(from, false, to, top);
  return compare(from, this.from) > 0 ? new Submap(from, false, to, top) : this;
 }

 @Override
 public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE
to) {
  if (top && bottom) return new Submap(from, false, to, false);
  if (! top) to = compare(to, this.to) < 0 ? to : this.to;
  if (! bottom) from = compare(from, this.from) > 0 ? from : this.from;
  if (! top && ! bottom && from == this.from && to == this.to) return this;
  return new Submap(from, false, to, false);
 }

 /** Locates the first entry.
  *
  * @return the first entry of this submap, or {@code null} if the submap is empty.
  */
 public RB_TREE_MAP.Entry KEY_VALUE_GENERIC firstEntry() {
  if (tree == null) return null;
  // If this submap goes to -infinity, we return the main map first entry; otherwise, we locate the start of the map.
  RB_TREE_MAP.Entry KEY_VALUE_GENERIC
e;
  if (bottom) e = firstEntry;
  else {
   e = locateKey(from);
   // If we find either the start or something greater we're OK.
   if (compare(e.key, from) < 0) e = e.next();
  }
  // Finally, if this submap doesn't go to infinity, we check that the resulting key isn't greater than the end.
  if (e == null || ! top && compare(e.key, to) >= 0) return null;
  return e;
 }

 /** Locates the last entry.
  *
  * @return the last entry of this submap, or {@code null} if the submap is empty.
  */
 public RB_TREE_MAP.Entry KEY_VALUE_GENERIC lastEntry() {
  if (tree == null) return null;
  // If this submap goes to infinity, we return the main map last entry; otherwise, we locate the end of the map.
  RB_TREE_MAP.Entry KEY_VALUE_GENERIC e;
  if (top) e = lastEntry;
  else {
```

```
    e = locateKey(to);
    // If we find something smaller than the end we're OK.
    if (compare(e.key, to) >= 0) e = e.prev();
   }
   // Finally, if this submap doesn't go to -infinity, we
  check that the resulting key isn't smaller than the start.
   if (e == null || ! bottom && compare(e.key, from) < 0) return null;
   return e;
  }

  @Override
  public KEY_GENERIC_TYPE FIRST_KEY() {
   RB_TREE_MAP.Entry KEY_VALUE_GENERIC e = firstEntry();
   if (e == null) throw new NoSuchElementException();
   return e.key;
  }

  @Override
  public KEY_GENERIC_TYPE LAST_KEY() {
   RB_TREE_MAP.Entry KEY_VALUE_GENERIC e = lastEntry();
   if (e == null) throw new NoSuchElementException();
   return e.key;
  }

  /** An iterator for subranges.
   *
   * <p>This class inherits from {@link TreeIterator}, but overrides the methods that
   * update the pointer after a {@link java.util.ListIterator#next()} or {@link java.util.ListIterator#previous()}. If we
 would
   * move out of the range of the submap we just overwrite the next or previous
   * entry with {@code null}.
   */
  private class SubmapIterator extends TreeIterator {
   SubmapIterator() {
    next = firstEntry();
   }

   SubmapIterator(final
  KEY_GENERIC_TYPE k) {
    this();

    if (next != null) {
     if (! bottom && compare(k, next.key) < 0) prev = null;
     else if (! top && compare(k, (prev = lastEntry()).key) >= 0) next = null;
     else {
      next = locateKey(k);

      if (compare(next.key, k) <= 0) {
```

```
      prev = next;
      next = next.next();
     }
     else prev = next.prev();
    }
   }
  }

  @Override
  void updatePrevious() {
   prev = prev.prev();
   if (! bottom && prev != null && RB_TREE_MAP.this.compare(prev.key, from) < 0) prev = null;
  }

  @Override
  void updateNext() {
   next = next.next();
   if (! top && next != null && RB_TREE_MAP.this.compare(next.key, to) >= 0) next = null;
  }
 }

 private class SubmapEntryIterator extends SubmapIterator implements ObjectListIterator<MAP.Entry
KEY_VALUE_GENERIC> {
  SubmapEntryIterator() {}

  SubmapEntryIterator(final KEY_GENERIC_TYPE k) {
   super(k);
  }

  @Override
  public MAP.Entry KEY_VALUE_GENERIC next()
 { return nextEntry(); }
  @Override
  public MAP.Entry KEY_VALUE_GENERIC previous() { return previousEntry(); }
 }


 /** An iterator on a subrange of keys.
  *
  * <p>This class can iterate in both directions on a subrange of the
  * keys of a threaded tree. We simply override the {@link
  * java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()} methods (and possibly their
  * type-specific counterparts) so that they return keys instead of
  * entries.
  */
 private final class SubmapKeyIterator extends SubmapIterator implements KEY_LIST_ITERATOR
KEY_GENERIC {
  public SubmapKeyIterator() { super(); }
```

```
  public SubmapKeyIterator(KEY_GENERIC_TYPE from) { super(from); }

  @Override
  public KEY_GENERIC_TYPE NEXT_KEY() { return nextEntry().key; }
  @Override
  public KEY_GENERIC_TYPE PREV_KEY() { return previousEntry().key; }
 };

 /** An iterator on a subrange of values.
  *
  * <p>This class can iterate in both directions on the values of a

* subrange of the keys of a threaded tree. We simply override the
  * {@link java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()} methods (and possibly their
  * type-specific counterparts) so that they return values instead of
  * entries.
  */
 private final class SubmapValueIterator extends SubmapIterator implements VALUE_LIST_ITERATOR
VALUE_GENERIC {
  @Override
  public VALUE_GENERIC_TYPE NEXT_VALUE() { return nextEntry().value; }
  @Override
  public VALUE_GENERIC_TYPE PREV_VALUE() { return previousEntry().value; }
 };
}


/** Returns a deep copy of this tree map.
 *
 * <p>This method performs a deep copy of this tree map; the data stored in the
 * set, however, is not cloned. Note that this makes a difference only for object keys.
 *
 * @return a deep copy of this tree map.
 */
@Override
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public RB_TREE_MAP KEY_VALUE_GENERIC clone() {
 RB_TREE_MAP KEY_VALUE_GENERIC c;
 try {
  c = (RB_TREE_MAP KEY_VALUE_GENERIC)super.clone();
 }
 catch(CloneNotSupportedException
cantHappen) {
  throw new InternalError();
 }

 c.keys = null;
 c.values = null;
```

```
  c.entries = null;
  c.allocatePaths();

  if (count != 0) {
   // Also this apparently unfathomable code is derived from GNU libavl.
   Entry KEY_VALUE_GENERIC e, p, q, rp = new Entry KEY_VALUE_GENERIC_DIAMOND(), rq = new Entry
KEY_VALUE_GENERIC_DIAMOND();

   p = rp;
   rp.left(tree);

   q = rq;
   rq.pred(null);

   while(true) {
    if (! p.pred()) {
     e = p.left.clone();
     e.pred(q.left);
     e.succ(q);
     q.left(e);

     p = p.left;
     q = q.left;
    }
    else {
     while(p.succ()) {
      p = p.right;

      if (p == null) {
       q.right = null;
       c.tree = rq.left;

       c.firstEntry = c.tree;
       while(c.firstEntry.left != null) c.firstEntry = c.firstEntry.left;
       c.lastEntry = c.tree;
       while(c.lastEntry.right != null) c.lastEntry = c.lastEntry.right;

       return c;
      }
      q = q.right;
     }

     p
 = p.right;
     q = q.right;
    }

    if (! p.succ()) {
```

```
    e = p.right.clone();
    e.succ(q.right);
    e.pred(q);
    q.right(e);
   }
  }
 }

 return c;
}


private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
 int n = count;
 EntryIterator i = new EntryIterator();
 Entry KEY_VALUE_GENERIC e;

 s.defaultWriteObject();

 while(n-- != 0) {
  e = i.nextEntry();
  s.WRITE_KEY(e.key);
  s.WRITE_VALUE(e.value);
 }
}


/** Reads the given number of entries from the input stream, returning the corresponding tree.
 *
 * @param s the input stream.
 * @param n the (positive) number of entries to read.
 * @param pred the entry containing the key that preceeds the first key in the tree.
 * @param succ the entry containing the key that follows the last key in the tree.
 */
 SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
 private Entry KEY_VALUE_GENERIC readTree(final java.io.ObjectInputStream s, final int n, final Entry
KEY_VALUE_GENERIC
 pred, final Entry KEY_VALUE_GENERIC succ) throws java.io.IOException, ClassNotFoundException {
  if (n == 1) {
   final Entry KEY_VALUE_GENERIC top = new Entry
KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY(), VALUE_GENERIC_CAST
s.READ_VALUE());
   top.pred(pred);
   top.succ(succ);
   top.black(true);

   return top;
  }
```

```
  if (n == 2) {
   /* We handle separately this case so that recursion will
    *always* be on nonempty subtrees. */
   final Entry KEY_VALUE_GENERIC top = new Entry
KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY(), VALUE_GENERIC_CAST
s.READ_VALUE());
   top.black(true);
   top.right(new Entry KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY(),
VALUE_GENERIC_CAST s.READ_VALUE()));
   top.right.pred(top);
   top.pred(pred);
   top.right.succ(succ);

   return top;
  }

  // The right subtree is the largest one.
  final int rightN = n / 2, leftN = n - rightN - 1;

  final Entry KEY_VALUE_GENERIC top = new Entry KEY_VALUE_GENERIC_DIAMOND();

  top.left(readTree(s,
 leftN, pred, top));

  top.key = KEY_GENERIC_CAST s.READ_KEY();
  top.value = VALUE_GENERIC_CAST s.READ_VALUE();
  top.black(true);

  top.right(readTree(s, rightN, top, succ));

  if (n + 2 == ((n + 2)  & -(n + 2))) top.right.black(false); // Quick test for determining whether n + 2 is a power of 2.

  return top;
 }

 private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
  s.defaultReadObject();
  /* The storedComparator is now correctly set, but we must restore
     on-the-fly the actualComparator. */
  setActualComparator();
  allocatePaths();

  if (count != 0) {
   tree = readTree(s, count, null, null);
   Entry KEY_VALUE_GENERIC e;

   e = tree;
```

```
  while(e.left() != null) e = e.left();
  firstEntry = e;

  e = tree;
  while(e.right() != null) e = e.right();
  lastEntry = e;
 }
}

#ifdef ASSERTS_CODE
 private void checkNodePath() {
  for(int i = nodePath.length; i-- != 0;) assert nodePath[i] ==
 null : i;
 }

 private static KEY_VALUE_GENERIC int checkTree(Entry KEY_VALUE_GENERIC e, int d, int D) {
  if (e == null) return 0;
  if (e.black()) d++;
  if (e.left() != null) D = checkTree(e.left(), d, D);
  if (e.right() != null) D = checkTree(e.right(), d, D);
  if (e.left() == null && e.right() == null) {
   if (D == -1) D = d;
   else if (D != d) throw new AssertionError("Mismatch between number of black nodes (" + D + " and " + d + ")");
  }
  return D;
 }
#endif



#ifdef TEST

 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#else
  return Integer.toBinaryString(r.nextInt());
#endif
 }

 private static VALUE_TYPE genValue() {
#if VALUE_CLASS_Byte || VALUE_CLASS_Short || VALUE_CLASS_Character
  return (VALUE_TYPE)(r.nextInt());
```

```
#elif
 VALUES_PRIMITIVE
  return r.NEXT_VALUE();
#elif !VALUE_CLASS_Reference || KEY_CLASS_Reference
  return Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }


 private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
 private static java.text.FieldPosition p = new java.text.FieldPosition(0);

 private static String format(double d) {
  StringBuffer s = new StringBuffer();
  return format.format(d, s, p).toString();
 }

 private static void speedTest(int n, boolean comp) {
  int i, j;
  RB_TREE_MAP m;
  java.util.TreeMap t;
  KEY_TYPE k[] = new KEY_TYPE[n];
  KEY_TYPE nk[] = new KEY_TYPE[n];
  VALUE_TYPE v[] = new VALUE_TYPE[n];
  long ms;

  for(i = 0; i < n; i++) {
   k[i] = genKey();
   nk[i] = genKey();
   v[i] = genValue();
  }

  double totPut = 0, totYes = 0, totNo = 0, totAddTo = 0, totIterFor = 0, totIterBack = 0, totRemYes = 0, d, dd, ddd;

  if (comp) { for(j = 0; j < 20; j++) {

   t = new java.util.TreeMap();



   /*
 We first add all pairs to t. */
   for(i = 0; i < n;  i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));

   /* Then we remove the first half and put it back. */
   for(i = 0; i < n/2;  i++) t.remove(KEY2OBJ(k[i]));
```

```
  ms = System.currentTimeMillis();
  for(i = 0; i < n/2;  i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));
  d = System.currentTimeMillis() - ms;

  /* Then we remove the other half and put it back again. */
  ms = System.currentTimeMillis();
  for(i = n/2; i < n;  i++) t.remove(KEY2OBJ(k[i]));
  dd = System.currentTimeMillis() - ms ;

  ms = System.currentTimeMillis();
  for(i = n/2; i < n;  i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));
  d += System.currentTimeMillis() - ms;
  if (j > 2) totPut += n/d;
  System.out.print("Add: " + format(n/d) +" K/s ");

  /* Then we remove again the first half. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n/2;  i++) t.remove(KEY2OBJ(k[i]));
  dd += System.currentTimeMillis() - ms ;
  if (j > 2) totRemYes +=
n/dd;
  System.out.print("RemYes: " + format(n/dd) +" K/s ");

  /* And then we put it back. */
  for(i = 0; i < n/2;  i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));

  #if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
  /* we perform n/2 addTo() operations with get then put */
  ms = System.currentTimeMillis();
  for(i = 0; i < n/2; i++) t.put(KEY2OBJ(k[i]), (VALUE_TYPE) ((VALUE_CLASS) t.get(KEY2OBJ(k[i])) + i));
  ddd = System.currentTimeMillis() - ms;
  if (j > 2) totAddTo += n/ddd;
  System.out.print("AddTo: " + format(n/ddd) +" K/s ");
  #endif

  /* We check for pairs in t. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) t.containsKey(KEY2OBJ(k[i]));
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totYes += d;
  System.out.print("Yes: " + format(d) +" K/s ");

  /* We check for pairs not in t. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) t.containsKey(KEY2OBJ(nk[i]));
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if
```

```
(j > 2) totNo += d;
  System.out.print("No: " + format(d) +" K/s ");

  /* We iterate on t. */
  ms = System.currentTimeMillis();
  for(Iterator it = t.entrySet().iterator(); it.hasNext(); it.next());
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totIterFor += d;
  System.out.print("IterFor: " + format(d) +" K/s ");

  System.out.println();
 }

 System.out.println();
 System.out.println("java.util Put: " + format(totPut/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s Yes:
" + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3))+ "K/s AddTo: " + format(totAddTo/(j-3)) + " K/s
IterFor: " + format(totIterFor/(j-3))  + " K/s");

 System.out.println();

 t = null;
 totPut = totYes = totNo = totIterFor = totIterBack = totRemYes = 0;

 }

 for(j = 0; j < 20; j++) {

  m = new RB_TREE_MAP();


  /* We first add all pairs to m. */
  for(i = 0; i < n;  i++) m.put(k[i], v[i]);

  /* Then we remove the first half and put it back. */
  for(i
= 0; i < n/2;  i++) m.remove(k[i]);

  ms = System.currentTimeMillis();
  for(i = 0; i < n/2;  i++) m.put(k[i], v[i]);
  d = System.currentTimeMillis() - ms;

  /* Then we remove the other half and put it back again. */
  ms = System.currentTimeMillis();
  for(i = n/2; i < n;  i++) m.remove(k[i]);
  dd = System.currentTimeMillis() - ms ;

  ms = System.currentTimeMillis();
  for(i = n/2; i < n;  i++) m.put(k[i], v[i]);
```

```
  d += System.currentTimeMillis() - ms;
  if (j > 2) totPut += n/d;
  System.out.print("Add: " + format(n/d) +" K/s ");

  /* Then we remove again the first half. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n/2;  i++) m.remove(k[i]);
  dd += System.currentTimeMillis() - ms ;
  if (j > 2) totRemYes += n/dd;
  System.out.print("RemYes: " + format(n/dd) +" K/s ");

  /* And then we put it back. */
  for(i = 0; i < n/2;  i++) m.put(k[i], v[i]);

  #if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
  /* we perform n/2 addTo() operations with get
then put */
  ms = System.currentTimeMillis();
  for(i = 0; i < n/2; i++) m.addTo(k[i], (VALUE_TYPE) i);
  ddd = System.currentTimeMillis() - ms;
  if (j > 2) totAddTo += n/ddd;
  System.out.print("AddTo: " + format(n/ddd) +" K/s ");
  #endif

  /* We check for pairs in m. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) m.containsKey(k[i]);
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totYes += d;
  System.out.print("Yes: " + format(d) +" K/s ");

  /* We check for pairs not in m. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) m.containsKey(nk[i]);
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totNo += d;
  System.out.print("No: " + format(d) +" K/s ");

  /* We iterate on m. */
  java.util.ListIterator it = (java.util.ListIterator)m.entrySet().iterator();
  ms = System.currentTimeMillis();
  for(; it.hasNext(); it.next());
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totIterFor
+= d;
  System.out.print("IterFor: " + format(d) +" K/s ");

  /* We iterate back on m. */
```

```
  ms = System.currentTimeMillis();
  for(; it.hasPrevious(); it.previous());
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totIterBack += d;
  System.out.print("IterBack: " + format(d) +" K/s ");

  System.out.println();
 }


 System.out.println();
 System.out.println("fastutil Put: " + format(totPut/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s Yes:
" + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3))+ "K/s AddTo: " + format(totAddTo/(j-3)) + " K/s
IterFor: " + format(totIterFor/(j-3))  + " K/s");

 System.out.println();

}


private static boolean valEquals(Object o1, Object o2) {
 return o1 == null ? o2 == null : o1.equals(o2);
}

private static void fatal(String msg) {
 System.out.println(msg);
 System.exit(1);
}

private static void ensure(boolean cond, String msg) {
 if (cond) return;
 fatal(msg);
}

private static Object[]
k, v, nk;
private static KEY_TYPE kt[];
private static KEY_TYPE nkt[];
private static VALUE_TYPE vt[];
private static RB_TREE_MAP topMap;

protected static void testMaps(SORTED_MAP m, SortedMap t, int n, int level) {
 long ms;
 boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement;
 Object rt = null, rm = null;

 if (level > 4) return;
```

```
  /* Now we check that both maps agree on first/last keys. */

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.firstKey();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.firstKey();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): firstKey() divergence at
start in NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  if (! mThrowsNoElement) ensure(t.firstKey().equals(m.firstKey()), "Error (" + level + ", " +
 seed + "): m and t differ at start on their first key (" + m.firstKey() + ", " + t.firstKey() +")");

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.lastKey();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.lastKey();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): lastKey() divergence at start
in NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");


  if (! mThrowsNoElement) ensure(t.lastKey().equals(m.lastKey()), "Error (" + level + ", " + seed + "): m and t differ
at start on their last key (" + m.lastKey() + ", " + t.lastKey() +")");


  /* Now we check that m and t are equal. */
  if (!m.equals(t) || ! t.equals(m)) System.err.println("m: " + m + " t: " + t);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
  ensure(t.equals(m), "Error
 (" + level + ", " + seed + "): ! t.equals(m) at start");



  /* Now we check that m actually holds that data. */
```

```
  for(Iterator i=t.entrySet().iterator(); i.hasNext();) {
   java.util.Map.Entry e = (java.util.Map.Entry)i.next();
   ensure(valEquals(e.getValue(), m.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after insertion (iterating on t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */
  for(Iterator i=m.entrySet().iterator(); i.hasNext();) {
   Entry e = (Entry)i.next();
   ensure(valEquals(e.getValue(), t.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after insertion (iterating on m)");
  }

  /* Now we check that m actually holds the same keys. */
  for(Iterator i=t.keySet().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(m.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key ("+o+") after insertion
(iterating on t)");
   ensure(m.keySet().contains(o),
 "Error (" + level + ", " + seed + "): m and t differ on a key ("+o+", in keySet()) after insertion (iterating on t)");
  }

  /* Now we check that m actually holds the same keys, but iterating on m. */
  for(Iterator i=m.keySet().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(t.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key after insertion (iterating on m)");
   ensure(t.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (in keySet()) after insertion
(iterating on m)");
  }


  /* Now we check that m actually hold the same values. */
  for(Iterator i=t.values().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(m.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after insertion (iterating on
t)");
   ensure(m.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
insertion (iterating
 on t)");
  }

  /* Now we check that m actually hold the same values, but iterating on m. */
  for(Iterator i=m.values().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(t.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after insertion (iterating on
m)");
   ensure(t.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
insertion (iterating on m)");
```

```
  }

  /* Now we check that inquiries about random data give the same answer in m and t. For
     m we use the polymorphic method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

   try {
    m.containsKey(KEY2OBJ(T));
   }
   catch (NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }

   try {
    t.containsKey(KEY2OBJ(T));
   }
   catch (NoSuchElementException e) { tThrowsNoElement
 = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): containsKey() divergence in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): containsKey() divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
   if (!mThrowsNoElement && !mThrowsIllegal) {
    ensure(m.containsKey(KEY2OBJ(T)) == t.containsKey(KEY2OBJ(T)), "Error (" + level + ", " + seed + "):
divergence in keys between t and m (polymorphic method)");

#if KEY_CLASS_Object && ! (VALUES_REFERENCE)
    if ((m.GET_VALUE(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
     t.get(KEY2OBJ(T)) != null &&
     ! VALUE2OBJ(m.GET_VALUE(T)).equals(t.get(KEY2OBJ(T))))
#else
     if ((m.get(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T))
 == null ? VALUE_NULL : VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
      t.get(KEY2OBJ(T)) != null &&
      ! m.get(KEY2OBJ(T)).equals(t.get(KEY2OBJ(T))))
#endif
      {
       System.out.println("Error (" + level + ", " + seed + "): divergence between t and m (polymorphic method)");
       System.exit(1);
      }
   }
  }
```

```
  /* Again, we check that inquiries about random data give the same answer in m and t, but
     for m we use the standard method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

   try {
    m.get(KEY2OBJ(T));
   }
   catch (NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }

   try {
    t.get(KEY2OBJ(T));
   }
   catch (NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }

   ensure(mThrowsNoElement == tThrowsNoElement,
 "Error (" + level + ", " + seed + "): get() divergence in NoSuchElementException for " + T + " (" +
mThrowsNoElement + ", " + tThrowsNoElement + ")");
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): get() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
   if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(m.get(KEY2OBJ(T)), t.get(KEY2OBJ(T))),
"Error (" + level + ", " + seed + "): divergence between t and m (standard method)");
  }

  /* Now we put and remove random data in m and t, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();
   VALUE_TYPE U = genValue();

   mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

   try {
    rm = m.put(KEY2OBJ(T), VALUE2OBJ(U));
   }
   catch (NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }

   try {
    rt
 = t.put(KEY2OBJ(T), VALUE2OBJ(U));
   }
```

```
  catch (NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }


  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): put() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): put() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(rm, rt), "Error (" + level + ", " + seed + "):
divergence in put() between t and m (" + rt + ", " + rm + ")");

  T = genKey();

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   rm = m.remove(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal
 = true; }

  try {
   rt = t.remove(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }


  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(rm, rt), "Error (" + level + ", " + seed + "):
divergence in remove() between t and m (" + rt + ", " + rm + ")");
 }

 ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal");
 ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal");

 /* Now we check that m actually holds
 the same data. */

 for(Iterator i=t.entrySet().iterator(); i.hasNext();) {
  java.util.Map.Entry e = (java.util.Map.Entry)i.next();
  ensure(valEquals(e.getValue(), m.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after removal (iterating on t)");
 }
```

```
  /* Now we check that m actually holds that data, but iterating on m. */

  for(Iterator i=m.entrySet().iterator(); i.hasNext();) {
   Entry e = (Entry)i.next();
   ensure(valEquals(e.getValue(), t.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after removal (iterating on m)");
  }

  /* Now we check that m actually holds the same keys. */

  for(Iterator i=t.keySet().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(m.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key ("+o+") after removal (iterating
on t)");
   ensure(m.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key ("+o+",
 in keySet()) after removal (iterating on t)");
  }

  /* Now we check that m actually holds the same keys, but iterating on m. */

  for(Iterator i=m.keySet().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(t.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key after removal (iterating on m)");
   ensure(t.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (in keySet()) after removal
(iterating on m)");
  }


  /* Now we check that m actually hold the same values. */

  for(Iterator i=t.values().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(m.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after removal (iterating on
t)");
   ensure(m.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
removal (iterating on t)");
  }

  /* Now we check that m actually hold the same values, but iterating on m. */

  for(Iterator
 i=m.values().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(t.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after removal (iterating on
m)");
   ensure(t.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after removal
(iterating on m)");
```

```
  }

  /* Now we check that both maps agree on first/last keys. */

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.firstKey();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.firstKey();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): firstKey() divergence in
NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  if (! mThrowsNoElement) ensure(t.firstKey().equals(m.firstKey()), "Error (" + level + ", " + seed + "): m and t differ
on their
 first key (" + m.firstKey() + ", " + t.firstKey() +")");

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.lastKey();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.lastKey();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): lastKey() divergence in
NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");

  if (! mThrowsNoElement) ensure(t.lastKey().equals(m.lastKey()), "Error (" + level + ", " + seed + "): m and t differ
on their last key (" + m.lastKey() + ", " + t.lastKey() +")");

  /* Now we check cloning. */

  if (level == 0) {
   ensure(m.equals(((RB_TREE_MAP)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
   ensure(((RB_TREE_MAP)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
   m = (RB_TREE_MAP)((RB_TREE_MAP)m).clone();
  }

  int
```

```
 h = m.hashCode();


  /* Now we save and read m. */

  SORTED_MAP m2 = null;

  try {
   java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
   java.io.OutputStream os = new java.io.FileOutputStream(ff);
   java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

   oos.writeObject(m);
   oos.close();

   java.io.InputStream is = new java.io.FileInputStream(ff);
   java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

   m2 = (SORTED_MAP)ois.readObject();
   ois.close();
   ff.delete();
  }
  catch(Exception e) {
   e.printStackTrace();
   System.exit(1);
  }

#if !VALUE_CLASS_Reference
  ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

  /* Now we check that m2 actually holds that data. */

  ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
  ensure(t.equals(m2), "Error
 (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
  /* Now we take out of m everything, and check that it is empty. */
#else
  m2.clear();
  m2.putAll(m);
#endif

  for(Iterator i=t.keySet().iterator(); i.hasNext();) m2.remove(i.next());

  ensure(m2.isEmpty(), "Error (" + level + ", " + seed + "): m2 is not empty (as it should be)");

  /* Now we play with iterators. */

  {
```

```
  java.util.ListIterator i, j;
  Map.Entry E, F;
  Object J;
  i = (java.util.ListIterator)m.entrySet().iterator();
  j = new java.util.LinkedList(t.entrySet()).listIterator();

  for(int k = 0; k < 2*n; k++) {
   ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext()");
   ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious()");

   if (r.nextFloat() < .8 && i.hasNext()) {
    ensure((E=(Entry)i.next()).getKey().equals(J = (F=(Map.Entry)j.next()).getKey()), "Error (" + level + ", " + seed +
"):
 divergence in next()");

    if (r.nextFloat() < 0.3) {
     i.remove();
     j.remove();
     t.remove(J);
    }
    else if (r.nextFloat() < 0.3) {
     Object U = VALUE2OBJ(genValue());
     E.setValue(U);
              t.put(F.getKey(), U);
    }
   }
   else if (r.nextFloat() < .2 && i.hasPrevious()) {
    ensure((E=(Entry)i.previous()).getKey().equals(J = (F=(Map.Entry)j.previous()).getKey()), "Error (" + level + ", " +
seed + "): divergence in previous()");

    if (r.nextFloat() < 0.3) {
     i.remove();
     j.remove();
     t.remove(J);
    }
    else if (r.nextFloat() < 0.3) {
     Object U = VALUE2OBJ(genValue());
     E.setValue(U);
              t.put(F.getKey(), U);
    }
   }

   ensure(i.nextIndex() == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
   ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

  }
 }
```

```
  {
  boolean badPrevious
 = false;
  Object previous = null;
  it.unimi.dsi.fastutil.BidirectionalIterator i;
  java.util.ListIterator j;
  Object I, J;
  KEY_TYPE from = genKey();
  j = new java.util.LinkedList(t.keySet()).listIterator();
  while(j.hasNext()) {
   Object k = j.next();
   if (((Comparable)k).compareTo(KEY2OBJ(from)) > 0) {
    badPrevious = true;
    j.previous();
    break;
   }
   previous = k;
  }

  i = (it.unimi.dsi.fastutil.BidirectionalIterator)((SORTED_SET)m.keySet()).iterator(from);

  for(int k = 0; k < 2*n; k++) {
   ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting
point " + from + ")");
   ensure(i.hasPrevious() == j.hasPrevious() || badPrevious && (i.hasPrevious() == (previous != null)), "Error (" +
level + ", " + seed + "): divergence in hasPrevious() (iterator with starting point " + from + ")" + badPrevious);

   if (r.nextFloat() < .8 && i.hasNext()) {
    ensure((I = i.next()).equals(J
 = j.next()), "Error (" + level + ", " + seed + "): divergence in next() (" + I + ", " + J + ", iterator with starting point "
+ from + ")");
    //System.err.println("Done next " + I + " " + J + "  " + badPrevious);

    badPrevious = false;

    if (r.nextFloat() < 0.5) {
     //System.err.println("Removing in next");
     i.remove();
     j.remove();
     t.remove(J);
    }
   }
   else if (!badPrevious && r.nextFloat() < .2 && i.hasPrevious()) {
    ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

    if (r.nextFloat() < 0.5) {
     //System.err.println("Removing in prev");
```

```
      i.remove();
      j.remove();
      t.remove(J);
     }
    }
   }

  }

  /* Now we check that m actually holds that data. */

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
  ensure(t.equals(m), "Error (" + level
 + ", " + seed + "): ! t.equals(m) after iteration");

  /* Now we select a pair of keys and create a submap. */

  if (! m.isEmpty()) {
   java.util.ListIterator i;
   Object start = m.firstKey(), end = m.firstKey();
   for(i = (java.util.ListIterator)m.keySet().iterator(); i.hasNext() && r.nextFloat() < .3; start = end = i.next());
   for(; i.hasNext() && r.nextFloat() < .95; end = i.next());

   //System.err.println("Checking subMap from " + start + " to " + end + " (level=" + (level+1) + ")...");
   testMaps((SORTED_MAP)m.subMap((KEY_CLASS) start, (KEY_CLASS)end), t.subMap(start, end), n, level +
1);

   ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after subMap");
   ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subMap");

   //System.err.println("Checking headMap to " + end + " (level=" + (level+1) + ")...");
   testMaps((SORTED_MAP)m.headMap((KEY_CLASS)end), t.headMap(end), n, level + 1);

   ensure(m.equals(t), "Error
(" + level + ", " + seed + "): ! m.equals(t) after headMap");
   ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after headMap");

   //System.err.println("Checking tailMap from " + start + " (level=" + (level+1) + ")...");
   testMaps((SORTED_MAP)m.tailMap((KEY_CLASS)start), t.tailMap(start), n, level + 1);

   ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after tailMap");
   ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after tailMap");
  }


 }
```

```
 private static void runTest(int n) {
  RB_TREE_MAP m = new RB_TREE_MAP();
  SortedMap t = new java.util.TreeMap();
  topMap = m;
  k = new Object[n];
  v = new Object[n];
  nk = new Object[n];
  kt = new KEY_TYPE[n];
  nkt = new KEY_TYPE[n];
  vt = new VALUE_TYPE[n];

  for(int i = 0; i < n; i++) {
#if KEY_CLASS_Object
   k[i] = kt[i] = genKey();
   nk[i] = nkt[i] = genKey();
#else
   k[i] = new KEY_CLASS(kt[i] = genKey());
   nk[i] = new KEY_CLASS(nkt[i] = genKey());
#endif
#if
 VALUES_REFERENCE
   v[i] = vt[i] = genValue();
#else
   v[i] = new VALUE_CLASS(vt[i] = genValue());
#endif
  }

  /* We add pairs to t. */
  for(int i = 0; i < n;  i++) t.put(k[i], v[i]);

  /* We add to m the same data */
  m.putAll(t);

  testMaps(m, t, n, 0);

  System.out.println("Test OK");
  return;
 }


 public static void main(String args[]) {
  int n  = Integer.parseInt(args[1]);
  if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

  try {
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
   else if ("test".equals(args[0])) runTest(n);
  } catch(Throwable e) {
```

```
   e.printStackTrace(System.err);
   System.err.println("seed: " + seed);
  }
 }

#endif


}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/RBTreeMap.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.NoSuchElementException;

import it.unimi.dsi.fastutil.PriorityQueue;

/** A type-specific {@link PriorityQueue}; provides some additional methods that use polymorphism to avoid (un)boxing.
*
* <p>Additionally, this interface strengthens {@link #comparator()}.
*/

public
 interface PRIORITY_QUEUE extends PriorityQueue<KEY_CLASS> {

 /** Enqueues a new element.
```

```
 * @see PriorityQueue#enqueue(Object)
 * @param x the element to enqueue.
 */

void enqueue(KEY_GENERIC_TYPE x);

/** Dequeues the {@linkplain #first() first} element from the queue.
 * @see #dequeue()
 * @return the dequeued element.
 * @throws NoSuchElementException if the queue is empty.
 */

KEY_GENERIC_TYPE DEQUEUE();

/** Returns the first element of the queue.
 * @see #first()
 * @return the first element.
 * @throws NoSuchElementException if the queue is empty.
 */

KEY_GENERIC_TYPE FIRST();

/** Returns the last element of the queue, that is, the element the would be dequeued last (optional operation).
 * <p>This default implementation just throws an {@link UnsupportedOperationException}.
 * @see #last()
 * @return the last element.
 * @throws NoSuchElementException if the queue is empty.
 */

default KEY_GENERIC_TYPE LAST() { throw new UnsupportedOperationException();
}

/** Returns the comparator associated with this priority queue, or null if it uses its elements' natural ordering.
 *
 * <p>Note that this specification strengthens the one given in {@link PriorityQueue#comparator()}.
 * @see PriorityQueue#comparator()
 * @return the comparator associated with this priority queue.
 */
@Override
KEY_COMPARATOR comparator();

/** {@inheritDoc}
 * <p>This default implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default void enqueue(final KEY_GENERIC_CLASS x) { enqueue(x.KEY_VALUE()); }
```

```
/** {@inheritDoc}
 * <p>This default implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS dequeue() { return KEY2OBJ(DEQUEUE()); }

/** {@inheritDoc}
 * <p>This default implementation
delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS first() { return KEY2OBJ(FIRST()); }

/** {@inheritDoc}
 * <p>This default implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS last() { return KEY2OBJ(LAST()); }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/PriorityQueue.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

/**  An abstract class facilitating the creation of type-specific {@linkplain java.util.ListIterator list iterators}.
```

```
*
* @deprecated As of fastutil 8 this class is no longer necessary, as its previous abstract
* methods are now default methods of the type-specific interface.
*/

@Deprecated
public
 abstract class KEY_ABSTRACT_LIST_ITERATOR KEY_GENERIC extends
KEY_ABSTRACT_BIDI_ITERATOR KEY_GENERIC implements KEY_LIST_ITERATOR KEY_GENERIC {
 protected KEY_ABSTRACT_LIST_ITERATOR() {}
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractListIterator.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2010-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import static it.unimi.dsi.fastutil.BigArrays.ensureOffsetLength;
import static it.unimi.dsi.fastutil.BigArrays.length;

import it.unimi.dsi.fastutil.BigArrays;

#if KEYS_REFERENCE
import it.unimi.dsi.fastutil.Stack;
#endif

import java.util.Iterator;
import java.util.Collection;
import java.util.NoSuchElementException;
```

```
import
 it.unimi.dsi.fastutil.BigList;
import it.unimi.dsi.fastutil.BigListIterator;

/**  An abstract class providing basic methods for big lists implementing a type-specific big list interface. */

public abstract class ABSTRACT_BIG_LIST KEY_GENERIC extends ABSTRACT_COLLECTION
KEY_GENERIC implements BIG_LIST KEY_GENERIC, STACK KEY_GENERIC {

 protected ABSTRACT_BIG_LIST() {}

 /** Ensures that the given index is nonnegative and not greater than this big-list size.
  *
  * @param index an index.
  * @throws IndexOutOfBoundsException if the given index is negative or greater than this big-list size.
  */
 protected void ensureIndex(final long index) {
  if (index < 0)  throw new IndexOutOfBoundsException("Index (" + index + ") is negative");
  if (index > size64()) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than list size (" +
(size64()) + ")");
 }

 /** Ensures that the given index is nonnegative and smaller than this big-list size.
  *

 * @param index an index.
  * @throws IndexOutOfBoundsException if the given index is negative or not smaller than this big-list size.
  */
 protected void ensureRestrictedIndex(final long index) {
  if (index < 0)  throw new IndexOutOfBoundsException("Index (" + index + ") is negative");
  if (index >= size64()) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list
size (" + (size64()) + ")");
 }

 /** {@inheritDoc}
  *
  * <p>This implementation always throws an {@link UnsupportedOperationException}.
  */
 @Override
 public void add(final long index, final KEY_GENERIC_TYPE k) {
  throw new UnsupportedOperationException();
 }

 /** {@inheritDoc}
  *
  * <p>This implementation delegates to the type-specific version of {@link BigList#add(long, Object)}.
  */
```

```
@Override
public boolean add(final KEY_GENERIC_TYPE k) {
 add(size64(), k);
 return true;
}

/** {@inheritDoc}
 *
 * <p>This implementation always throws an {@link UnsupportedOperationException}.

*/
@Override
public KEY_GENERIC_TYPE REMOVE_KEY(long i) {
 throw new UnsupportedOperationException();
}

/** {@inheritDoc}
 *
 * <p>This implementation always throws an {@link UnsupportedOperationException}.
 */
@Override
public KEY_GENERIC_TYPE set(final long index, final KEY_GENERIC_TYPE k) {
 throw new UnsupportedOperationException();
}

/** Adds all of the elements in the specified collection to this list (optional operation). */
@Override
public boolean addAll(long index, final Collection<? extends KEY_GENERIC_CLASS> c) {
 ensureIndex(index);
 final Iterator<? extends KEY_GENERIC_CLASS> i = c.iterator();
 final boolean retVal = i.hasNext();
 while(i.hasNext()) add(index++, i.next());
 return retVal;
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link BigList#addAll(long, Collection)}.
 */
@Override
public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) {
 return addAll(size64(),
c);
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to {@link #listIterator()}.
```

```
 */
@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC iterator() {
 return listIterator();
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to {@link BigList#listIterator(long) listIterator(0)}.
 */
@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator() {
 return listIterator(0L);
}

/** {@inheritDoc}
 * <p>This implementation is based on the random-access methods. */
@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(final long index) {
 ensureIndex(index);

 return new KEY_BIG_LIST_ITERATOR KEY_GENERIC() {
   long pos = index, last = -1;

   @Override
   public boolean hasNext() { return pos < ABSTRACT_BIG_LIST.this.size64(); }

   @Override
   public boolean hasPrevious() { return pos > 0; }

   @Override
   public KEY_GENERIC_TYPE NEXT_KEY() { if (! hasNext()) throw new NoSuchElementException(); return
ABSTRACT_BIG_LIST.this.GET_KEY(last
 = pos++); }

   @Override
   public KEY_GENERIC_TYPE PREV_KEY() { if (! hasPrevious()) throw new NoSuchElementException(); return
ABSTRACT_BIG_LIST.this.GET_KEY(last = --pos); }

   @Override
   public long nextIndex() { return pos; }

   @Override
   public long previousIndex() { return pos - 1; }

   @Override
   public void add(KEY_GENERIC_TYPE k) {
    ABSTRACT_BIG_LIST.this.add(pos++, k);
```

```
    last = -1;
   }

   @Override
   public void set(KEY_GENERIC_TYPE k) {
    if (last == -1) throw new IllegalStateException();
    ABSTRACT_BIG_LIST.this.set(last, k);
   }

   @Override
   public void remove() {
    if (last == -1) throw new IllegalStateException();
    ABSTRACT_BIG_LIST.this.REMOVE_KEY(last);
    /* If the last operation was a next(), we are removing an element *before* us, and we must decrease pos
correspondingly. */
    if (last < pos) pos--;
    last = -1;
   }
  };
 }


 /** Returns true if this list contains the specified element.

 * <p>This implementation delegates to {@code indexOf()}.
  * @see BigList#contains(Object)
  */
 @Override
 public boolean contains(final KEY_TYPE k) { return indexOf(k) >= 0; }

 @Override
 public long indexOf(final KEY_TYPE k) {
  final KEY_BIG_LIST_ITERATOR KEY_GENERIC i = listIterator();
  KEY_GENERIC_TYPE e;
  while(i.hasNext()) {
   e = i.NEXT_KEY();
   if (KEY_EQUALS(k, e)) return i.previousIndex();
  }
  return -1;
 }

 @Override
 public long lastIndexOf(final KEY_TYPE k) {
  KEY_BIG_LIST_ITERATOR KEY_GENERIC i = listIterator(size64());
  KEY_GENERIC_TYPE e;
  while(i.hasPrevious()) {
   e = i.PREV_KEY();
   if (KEY_EQUALS(k, e)) return i.nextIndex();
```

```
 }
 return -1;
}

@Override
public void size(final long size) {
 long i = size64();
 if (size > i) while(i++ < size) add(KEY_NULL);
 else while(i-- != size) remove(i);
}

@Override
public BIG_LIST KEY_GENERIC subList(final long from, final long to) {
 ensureIndex(from);
 ensureIndex(to);
 if (from > to) throw new IndexOutOfBoundsException("Start
index (" + from + ") is greater than end index (" + to + ")");

 return new SUBLIST KEY_GENERIC_DIAMOND(this, from, to);
}

/** {@inheritDoc}
 *
 * <p>This is a trivial iterator-based implementation. It is expected that
 * implementations will override this method with a more optimized version.
 */
@Override
public void removeElements(final long from, final long to) {
 ensureIndex(to);
 KEY_BIG_LIST_ITERATOR KEY_GENERIC i = listIterator(from);
 long n = to - from;
 if (n < 0) throw new IllegalArgumentException("Start index (" + from + ") is greater than end index (" + to + ")");
 while(n-- != 0) {
  i.NEXT_KEY();
  i.remove();
 }
}

/** {@inheritDoc}
 *
 * <p>This is a trivial iterator-based implementation. It is expected that
 * implementations will override this method with a more optimized version.
 */
@Override
public void addElements(long index, final KEY_GENERIC_TYPE a[][], long offset, long length) {
 ensureIndex(index);
 ensureOffsetLength(a,
```

```
offset, length);
 while(length-- != 0) add(index++, BigArrays.get(a, offset++));
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the analogous method for big-array fragments.
 */
@Override
public void addElements(final long index, final KEY_GENERIC_TYPE a[][]) {
 addElements(index, a, 0, length(a));
}

/** {@inheritDoc}
 *
 * <p>This is a trivial iterator-based implementation. It is expected that
 * implementations will override this method with a more optimized version.
 */
@Override
public void getElements(final long from, final KEY_TYPE a[][], long offset, long length) {
 KEY_BIG_LIST_ITERATOR KEY_GENERIC i = listIterator(from);
 ensureOffsetLength(a, offset, length);
 if (from + length > size64()) throw new IndexOutOfBoundsException("End index (" + (from + length) + ") is
greater than list size (" + size64() + ")");
 while(length-- != 0) BigArrays.set(a, offset++, i.NEXT_KEY());
}

/** {@inheritDoc}
 * <p>This implementation
delegates to {@link #removeElements(long, long)}.*/
@Override
public void clear() {
 removeElements(0, size64());
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to {@link #size64()}.
 * @deprecated Please use {@link #size64()} instead. */
@Override
@Deprecated
public int size() {
 return (int)Math.min(Integer.MAX_VALUE, size64());
}

#if ! KEY_CLASS_Reference
 private boolean valEquals(final Object a, final Object b) { return a == null ? b == null : a.equals(b); }
#endif
```

```
 /** Returns the hash code for this big list, which is identical to {@link java.util.List#hashCode()}.
  *
  * @return the hash code for this big list.
  */
 @Override
 public int hashCode() {
  KEY_ITERATOR KEY_GENERIC i = iterator();
  int h = 1;
  long s = size64();
  while (s-- != 0) {
   KEY_GENERIC_TYPE k = i.NEXT_KEY();
   h = 31 * h + KEY2JAVAHASH(k);
  }
  return h;
 }

 @Override
 public boolean equals(final Object o) {
  if (o == this) return true;
  if (! (o instanceof BigList))
 return false;
  final BigList<?> l = (BigList<?>)o;
  long s = size64();
  if (s != l.size64()) return false;

#if KEYS_PRIMITIVE
  if (l instanceof BIG_LIST) {
   final KEY_BIG_LIST_ITERATOR KEY_GENERIC i1 = listIterator(), i2 = ((BIG_LIST
KEY_GENERIC)l).listIterator();
   while(s-- != 0) if (i1.NEXT_KEY() != i2.NEXT_KEY()) return false;
   return true;
  }
#endif

  final BigListIterator<?> i1 = listIterator(), i2 = l.listIterator();

#if KEY_CLASS_Reference
  while(s-- !=  0) if (i1.next() != i2.next()) return false;
#else
  while(s-- !=  0) if (! valEquals(i1.next(), i2.next())) return false;
#endif
  return true;
 }

#if ! KEY_CLASS_Reference
 /** Compares this big list to another object. If the
  * argument is a {@link BigList}, this method performs a lexicographical comparison; otherwise,
```

```
 * it throws a {@code ClassCastException}.
 *
 * @param l a big list.
 * @return if the argument is a {@link BigList}, a negative integer,
 * zero, or a positive integer as this list is
lexicographically less than, equal
 * to, or greater than the argument.
 * @throws ClassCastException if the argument is not a big list.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public int compareTo(final BigList<? extends KEY_GENERIC_CLASS> l) {
 if (l == this) return 0;

 if (l instanceof BIG_LIST) {

  final KEY_BIG_LIST_ITERATOR KEY_GENERIC i1 = listIterator(), i2 = ((BIG_LIST
KEY_GENERIC)l).listIterator();
  int r;
  KEY_GENERIC_TYPE e1, e2;

  while(i1.hasNext() && i2.hasNext()) {
   e1 = i1.NEXT_KEY();
   e2 = i2.NEXT_KEY();
   if ((r = KEY_CMP(e1, e2)) != 0) return r;
  }
  return i2.hasNext() ? -1 : (i1.hasNext() ? 1 : 0);
 }

 BigListIterator<? extends KEY_GENERIC_CLASS> i1 = listIterator(), i2 = l.listIterator();
 int r;

 while(i1.hasNext() && i2.hasNext()) {
  if ((r = ((Comparable<? super KEY_GENERIC_CLASS>)i1.next()).compareTo(i2.next())) != 0) return r;
 }
 return i2.hasNext() ? -1 : (i1.hasNext() ? 1 : 0);
}
#endif

@Override
public void
push(KEY_GENERIC_TYPE o) {
 add(o);
}

@Override
public KEY_GENERIC_TYPE POP() {
```

```
 if (isEmpty()) throw new NoSuchElementException();
 return REMOVE_KEY(size64() - 1);
}

@Override
public KEY_GENERIC_TYPE TOP() {
 if (isEmpty()) throw new NoSuchElementException();
 return GET_KEY(size64() - 1);
}

@Override
public KEY_GENERIC_TYPE PEEK(int i) {
 return GET_KEY(size64() - 1 - i);
}


#if KEYS_PRIMITIVE
/** Removes a single instance of the specified element from this collection, if it is present (optional operation).
 * <p>This implementation delegates to {@code indexOf()}.
 * @see BigList#remove(Object)
 */
@Override
public boolean rem(KEY_TYPE k) {
 long index = indexOf(k);
 if (index == -1) return false;
 REMOVE_KEY(index);
 return true;
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link #addAll(long, Collection)}.
 */
@Override
public boolean addAll(final long index, final COLLECTION c) { return addAll(index,
(Collection<? extends KEY_CLASS>)c); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link #addAll(long, Collection)}.
 */
@Override
public boolean addAll(final long index, final BIG_LIST l) { return addAll(index, (COLLECTION)l); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link #addAll(long, Collection)}.
 */
```

```
@Override
public boolean addAll(final COLLECTION c) { return addAll(size64(), c); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific list version of {@link #addAll(long, Collection)}.
 */
@Override
public boolean addAll(final BIG_LIST l) { return addAll(size64(), l); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public void add(final long index, final KEY_CLASS ok) { add(index,
ok.KEY_VALUE()); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public KEY_CLASS set(final long index, final KEY_CLASS ok) { return KEY2OBJ(set(index,
ok.KEY_VALUE())); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public KEY_CLASS get(final long index) { return KEY2OBJ(GET_KEY(index)); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public long indexOf(final Object ok) { return indexOf(KEY_OBJ2TYPE(ok)); }
```

```
/** {@inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific
method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public long lastIndexOf(final Object ok) { return lastIndexOf(KEY_OBJ2TYPE(ok)); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public KEY_CLASS remove(final long index) { return KEY2OBJ(REMOVE_KEY(index)); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public void push(KEY_CLASS o) { push(o.KEY_VALUE()); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public KEY_CLASS pop() { return KEY_CLASS.valueOf(POP());
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public KEY_CLASS top() { return KEY_CLASS.valueOf(TOP()); }

/** {@inheritDoc}
```

```
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public KEY_CLASS peek(int i) { return KEY_CLASS.valueOf(PEEK(i)); }

#endif

@Override
public String toString() {
 final StringBuilder s = new StringBuilder();
 final KEY_ITERATOR KEY_GENERIC i = iterator();
 long n = size64();
 KEY_GENERIC_TYPE k;
 boolean first = true;

 s.append("[");

 while(n-- != 0) {
  if (first) first = false;
  else s.append(", ");
  k = i.NEXT_KEY();
#if KEYS_REFERENCE
  if (this == k) s.append("(this big list)"); else
#endif
   s.append(String.valueOf(k));
 }

 s.append("]");
 return
s.toString();
}


/** A class implementing a sublist view. */
public static class SUBLIST KEY_GENERIC extends ABSTRACT_BIG_LIST KEY_GENERIC implements
java.io.Serializable {
 private static final long serialVersionUID = -7046029254386353129L;
 /** The list this sublist restricts. */
 protected final BIG_LIST KEY_GENERIC l;
 /** Initial (inclusive) index of this sublist. */
 protected final long from;
 /** Final (exclusive) index of this sublist. */
 protected long to;

 public SUBLIST(final BIG_LIST KEY_GENERIC l, final long from, final long to) {
```

```
 this.l = l;
 this.from = from;
 this.to = to;
}

private boolean assertRange() {
 assert from <= l.size64();
 assert to <= l.size64();
 assert to >= from;
 return true;
}

@Override
public boolean add(final KEY_GENERIC_TYPE k) {
 l.add(to, k);
 to++;
 assert assertRange();
 return true;
}

@Override
public void add(final long index, final KEY_GENERIC_TYPE
k) {
 ensureIndex(index);
 l.add(from + index, k);
 to++;
 assert assertRange();
}

@Override
public boolean addAll(final long index, final Collection<? extends KEY_GENERIC_CLASS> c) {
 ensureIndex(index);
 to += c.size();
 return l.addAll(from + index, c);
}

@Override
public KEY_GENERIC_TYPE GET_KEY(long index) {
 ensureRestrictedIndex(index);
 return l.GET_KEY(from + index);
}

@Override
public KEY_GENERIC_TYPE REMOVE_KEY(long index) {
 ensureRestrictedIndex(index);
 to--;
 return l.REMOVE_KEY(from + index);
}
```

```
@Override
public KEY_GENERIC_TYPE set(long index, KEY_GENERIC_TYPE k) {
 ensureRestrictedIndex(index);
 return l.set(from + index, k);
}

@Override
public long size64() { return to - from; }

@Override
public void getElements(final long from, final KEY_TYPE[][] a, final long offset, final long length) {
 ensureIndex(from);
 if (from + length > size64())  throw new IndexOutOfBoundsException("End index (" + from
+ length + ") is greater than list size (" + size64() + ")");
 l.getElements(this.from + from, a, offset, length);
}

@Override
public void removeElements(final long from, final long to) {
 ensureIndex(from);
 ensureIndex(to);
 l.removeElements(this.from + from, this.from + to);
 this.to -= (to - from);
 assert assertRange();
}

@Override
public void addElements(final long index, final KEY_GENERIC_TYPE a[][], long offset, long length) {
 ensureIndex(index);
 l.addElements(this.from + index, a, offset, length);
 this.to += length;
 assert assertRange();
}

@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(final long index) {
 ensureIndex(index);

 return new KEY_BIG_LIST_ITERATOR KEY_GENERIC() {
  long pos = index, last = -1;

  @Override
  public boolean hasNext() { return pos < size64(); }
  @Override
  public boolean hasPrevious() { return pos > 0; }
  @Override
  public KEY_GENERIC_TYPE NEXT_KEY() { if (! hasNext())
```

```
 throw new NoSuchElementException(); return l.GET_KEY(from + (last = pos++)); }
    @Override
    public KEY_GENERIC_TYPE PREV_KEY() { if (! hasPrevious()) throw new NoSuchElementException(); return
l.GET_KEY(from + (last = --pos)); }
    @Override
    public long nextIndex() { return pos; }
    @Override
    public long previousIndex() { return pos - 1; }
    @Override
    public void add(KEY_GENERIC_TYPE k) {
     if (last == -1) throw new IllegalStateException();
     SUBLIST.this.add(pos++, k);
     last = -1;
     assert assertRange();
    }
    @Override
    public void set(KEY_GENERIC_TYPE k) {
     if (last == -1) throw new IllegalStateException();
     SUBLIST.this.set(last, k);
    }
    @Override
    public void remove() {
     if (last == -1) throw new IllegalStateException();
     SUBLIST.this.REMOVE_KEY(last);
     /* If the last operation was a next(), we are removing an element *before* us, and we must decrease pos
correspondingly. */
     if (last < pos) pos--;
     last
 = -1;
     assert assertRange();
    }
   };
  }

  @Override
  public BIG_LIST KEY_GENERIC subList(final long from, final long to) {
   ensureIndex(from);
   ensureIndex(to);
   if (from > to) throw new IllegalArgumentException("Start index (" + from + ") is greater than end index (" + to +
")");

   return new SUBLIST KEY_GENERIC_DIAMOND(this, from, to);
  }

#if KEYS_PRIMITIVE
  @Override
  public boolean rem(KEY_TYPE k) {
   long index = indexOf(k);
```

```
  if (index == -1) return false;
  to--;
  l.REMOVE_KEY(from + index);
  assert assertRange();
  return true;
 }

 @Override
 public boolean addAll(final long index, final COLLECTION c) {
  ensureIndex(index);
  return super.addAll(index, c);
 }

 @Override
 public boolean addAll(final long index, final BIG_LIST l) {
  ensureIndex(index);
  return super.addAll(index, l);
 }
#endif
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractBigList.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

#if KEY_CLASS_Object
import java.util.Comparator;
```

```
#endif

/** A class providing static methods and objects that do useful things with heaps.
*
* <p>The static methods of this class allow to treat arrays as 0-based heaps. They
* are used in the implementation of heap-based queues, but
 they may be also used
* directly.
*
*/

public final class HEAPS {

 private HEAPS() {}

 /** Moves the given element down into the heap until it reaches the lowest possible position.
  *
  * @param heap the heap (starting at 0).
  * @param size the number of elements in the heap.
  * @param i the index of the element that must be moved down.
  * @param c a type-specific comparator, or {@code null} for the natural order.
  * @return the new position of the element of index {@code i}.
  */

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public static KEY_GENERIC int downHeap(final KEY_GENERIC_TYPE[] heap, final int size, int i, final
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
  assert i < size;

  final KEY_GENERIC_TYPE e = heap[i];
  int child;

  if (c == null)
   while ((child = (i << 1) + 1) < size) {
    KEY_GENERIC_TYPE t = heap[child];
    final int right = child + 1;
    if (right < size && KEY_LESS(heap[right], t)) t = heap[child = right];
    if (KEY_LESSEQ(e, t)) break;
    heap[i] = t;
    i
 = child;
   }
  else
   while ((child = (i << 1) + 1) < size) {
    KEY_GENERIC_TYPE t = heap[child];
    final int right = child + 1;
    if (right < size && c.compare(heap[right], t) < 0) t = heap[child = right];
    if (c.compare(e, t) <= 0) break;
```

```
  heap[i] = t;
  i = child;
  }

 heap[i] = e;

 return i;
}

/** Moves the given element up in the heap until it reaches the highest possible position.
 *
 * @param heap the heap (starting at 0).
 * @param size the number of elements in the heap.
 * @param i the index of the element that must be moved up.
 * @param c a type-specific comparator, or {@code null} for the natural order.
 * @return the new position of the element of index {@code i}.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC int upHeap(final KEY_GENERIC_TYPE[] heap, final int size, int i, final
KEY_COMPARATOR KEY_GENERIC c) {
 assert i < size;

 final KEY_GENERIC_TYPE e = heap[i];

 if (c == null)
  while (i != 0) {
   final int
parent = (i - 1) >>> 1;
   final KEY_GENERIC_TYPE t = heap[parent];
   if (KEY_LESSEQ(t, e)) break;
   heap[i] = t;
   i = parent;
  }
 else
  while (i != 0) {
   final int parent = (i - 1) >>> 1;
   final KEY_GENERIC_TYPE t = heap[parent];
   if (c.compare(t, e) <= 0) break;
   heap[i] = t;
   i = parent;
  }

 heap[i] = e;

 return i;
}
```

```
 /** Makes an array into a heap.
  *
  * @param heap the heap (starting at 0).
  * @param size the number of elements in the heap.
  * @param c a type-specific comparator, or {@code null} for the natural order.
  */

 public static KEY_GENERIC void makeHeap(final KEY_GENERIC_TYPE[] heap, final int size, final
KEY_COMPARATOR KEY_GENERIC c) {
  int i = size >>> 1;
  while(i-- != 0) downHeap(heap, size, i, c);
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Heaps.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/AbstractStack.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/HashCommon.java
*
 /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/Arrays.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/Stack.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/BidirectionalIterator.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-

gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/Function.java
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/it/unimi/dsi/fastutil/Hash.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.Collection;
import java.util.Comparator;
import java.util.Iterator;
import java.util.SortedSet;
import java.util.NoSuchElementException;

/** A type-specific AVL tree set with a fast, small-footprint implementation.
*
* <p>The iterators provided by this class are type-specific {@link
 * it.unimi.dsi.fastutil.BidirectionalIterator bidirectional iterators}.
* Moreover, the iterator returned by {@code iterator()} can be safely cast
* to a type-specific {@linkplain java.util.ListIterator list iterator}.
*/

public class AVL_TREE_SET KEY_GENERIC extends ABSTRACT_SORTED_SET KEY_GENERIC
implements java.io.Serializable, Cloneable, SORTED_SET KEY_GENERIC {

 /** A reference to the root entry. */
 protected transient Entry KEY_GENERIC tree;

 /** Number of elements in this set. */
 protected int count;

 /** The entry of the first element of this set. */
```

```
 protected transient Entry KEY_GENERIC firstEntry;

 /** The entry of the last element of this set. */
 protected transient Entry KEY_GENERIC lastEntry;

 /** This set's comparator, as provided in the constructor. */
 protected Comparator<? super KEY_GENERIC_CLASS> storedComparator;

 /** This set's actual comparator; it may differ from {@link #storedComparator} because it is
  always a type-specific comparator, so it
 could be derived from the former by wrapping. */
 protected transient KEY_COMPARATOR KEY_SUPER_GENERIC actualComparator;

 private static final long serialVersionUID = -7046029254386353130L;

 {
  allocatePaths();
 }

 /** Creates a new empty tree set.
  */

 public AVL_TREE_SET() {
  tree = null;
  count = 0;
 }

 /** Generates the comparator that will be actually used.
  *
  * <p>When a given {@link Comparator} is specified and stored in {@link
  * #storedComparator}, we must check whether it is type-specific.  If it is
  * so, we can used directly, and we store it in {@link #actualComparator}. Otherwise,
  * we adapt it using a helper static method.
  */
 private void setActualComparator() {
#if KEY_CLASS_Object
  actualComparator = storedComparator;
#else
  actualComparator = COMPARATORS.AS_KEY_COMPARATOR(storedComparator);
#endif
 }


 /** Creates a new empty tree set with the given comparator.
  *
  * @param c a {@link Comparator} (even better, a type-specific comparator).
  */
```

```
public
AVL_TREE_SET(final Comparator<? super KEY_GENERIC_CLASS> c) {
 this();
 storedComparator = c;
 setActualComparator();
}


/** Creates a new tree set copying a given set.
 *
 * @param c a collection to be copied into the new tree set.
 */

public AVL_TREE_SET(final Collection<? extends KEY_GENERIC_CLASS> c) {
 this();
 addAll(c);
}

/** Creates a new tree set copying a given sorted set (and its {@link Comparator}).
 *
 * @param s a {@link SortedSet} to be copied into the new tree set.
 */

public AVL_TREE_SET(final SortedSet <KEY_GENERIC_CLASS> s) {
 this(s.comparator());
 addAll(s);
}

/** Creates a new tree set copying a given type-specific collection.
 *
 * @param c a type-specific collection to be copied into the new tree set.
 */

public AVL_TREE_SET(final COLLECTION KEY_EXTENDS_GENERIC c) {
 this();
 addAll(c);
}

/** Creates a new tree set copying a given type-specific sorted set (and its {@link Comparator}).
 *
 * @param s a type-specific sorted
set to be copied into the new tree set.
 */

public AVL_TREE_SET(final SORTED_SET KEY_GENERIC s) {
 this(s.comparator());
 addAll(s);
}
```

```
 /** Creates a new tree set using elements provided by a type-specific iterator.
  *
  * @param i a type-specific iterator whose elements will fill the set.
  */

 public AVL_TREE_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i) {
  while(i.hasNext()) add(i.NEXT_KEY());
 }

#if KEYS_PRIMITIVE

 /** Creates a new tree set using elements provided by an iterator.
  *
  * @param i an iterator whose elements will fill the set.
  */

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public AVL_TREE_SET(final Iterator<?> i) {
  this(ITERATORS.AS_KEY_ITERATOR(i));
 }

#endif

 /** Creates a new tree set and fills it with the elements of a given array using a given {@link Comparator}.
  *
  * @param a an array whose elements will be used to fill the set.
  * @param offset the first element to use.
  * @param length the number of elements to use.
  * @param c a {@link
 Comparator} (even better, a type-specific comparator).
  */

 public AVL_TREE_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length, final Comparator<?
super KEY_GENERIC_CLASS> c) {
  this(c);
  ARRAYS.ensureOffsetLength(a, offset, length);
  for(int i = 0; i < length; i++) add(a[offset + i]);
 }


 /** Creates a new tree set and fills it with the elements of a given array.
  *
  * @param a an array whose elements will be used to fill the set.
  * @param offset the first element to use.
  * @param length the number of elements to use.
  */
```

```
public AVL_TREE_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length) {
 this(a, offset, length, null);
}


/** Creates a new tree set copying the elements of an array.
 *
 * @param a an array to be copied into the new tree set.
 */

public AVL_TREE_SET(final KEY_GENERIC_TYPE[] a) {
 this();
 int i = a.length;
 while(i-- != 0) add(a[i]);
}


/** Creates a new tree set copying the elements of an array using
a given {@link Comparator}.
 *
 * @param a an array to be copied into the new tree set.
 * @param c a {@link Comparator} (even better, a type-specific comparator).
 */

public AVL_TREE_SET(final KEY_GENERIC_TYPE[] a, final Comparator<? super KEY_GENERIC_CLASS>
c) {
 this(c);
 int i = a.length;
 while(i-- != 0) add(a[i]);
}




/*
 * The following methods implements some basic building blocks used by
 * all accessors.  They are (and should be maintained) identical to those used in AVLTreeMap.drv.
 *
 * The add()/remove() code is derived from Ben Pfaff's GNU libavl
 * (http://www.msu.edu/~pfaffben/avl/). If you want to understand what's
 * going on, you should have a look at the literate code contained therein
 * first.
 */


/** Compares two keys in the right way.
 *
```

```
 * <p>This method uses the {@link #actualComparator} if it is non-{@code null}.
 * Otherwise, it resorts to primitive type comparisons or to {@link Comparable#compareTo(Object) compareTo()}.
 *
 *
@param k1 the first key.
 * @param k2 the second key.
 * @return a number smaller than, equal to or greater than 0, as usual
 * (i.e., when k1 &lt; k2, k1 = k2 or k1 &gt; k2, respectively).
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
final int compare(final KEY_GENERIC_TYPE k1, final KEY_GENERIC_TYPE k2) {
 return actualComparator == null ? KEY_CMP(k1, k2) : actualComparator.compare(k1, k2);
}



/** Returns the entry corresponding to the given key, if it is in the tree; {@code null}, otherwise.
 *
 * @param k the key to search for.
 * @return the corresponding entry, or {@code null} if no entry with the given key exists.
 */

private Entry KEY_GENERIC findKey(final KEY_GENERIC_TYPE k) {
 Entry KEY_GENERIC e = tree;
 int cmp;

 while (e != null && (cmp = compare(k, e.key)) != 0)
  e = cmp < 0 ? e.left() : e.right();

 return e;
}

/** Locates a key.
 *
 * @param k a key.
 * @return the last entry on a search for the given key; this will be
 * the given key, if it present;
otherwise, it will be either the smallest greater key or the greatest smaller key.
 */

final Entry KEY_GENERIC locateKey(final KEY_GENERIC_TYPE k) {
 Entry KEY_GENERIC e = tree, last = tree;
 int cmp = 0;

 while (e != null && (cmp = compare(k, e.key)) != 0) {
  last = e;
  e = cmp < 0 ? e.left() : e.right();
```

```
 }

 return cmp == 0 ? e : last;
}


/** This vector remembers the path followed during the current insertion. It suffices for
 about 2<sup>32</sup> entries. */
private transient boolean dirPath[];

private void allocatePaths() {
 dirPath = new boolean[48];
}


@Override
public boolean add(final KEY_GENERIC_TYPE k) {

 if (tree == null) { // The case of the empty tree is treated separately.
  count++;
  tree = lastEntry = firstEntry = new Entry KEY_GENERIC_DIAMOND(k);
 }
 else {
  Entry KEY_GENERIC p = tree, q = null, y = tree, z = null, e = null, w = null;
  int cmp, i = 0;

  while(true) {
   if ((cmp = compare(k, p.key)) == 0) return false;

   if
(p.balance() != 0) {
    i = 0;
    z = q;
    y = p;
   }

   if (dirPath[i++] = cmp > 0) {
    if (p.succ()) {
     count++;
     e = new Entry KEY_GENERIC_DIAMOND(k);

     if (p.right == null) lastEntry = e;

     e.left = p;
     e.right = p.right;

     p.right(e);
```

```
   break;
  }

  q = p;
  p = p.right;
 }
 else {
  if (p.pred()) {
   count++;
   e = new Entry KEY_GENERIC_DIAMOND(k);

   if (p.left == null) firstEntry = e;

   e.right = p;
   e.left = p.left;

   p.left(e);

   break;
  }

  q = p;
  p = p.left;
 }
}

p = y;
i = 0;

while(p != e) {
 if (dirPath[i]) p.incBalance();
 else p.decBalance();

 p = dirPath[i++] ? p.right : p.left;
}

if (y.balance() == -2) {
 Entry KEY_GENERIC x = y.left;

 if (x.balance() == -1) {
  w = x;
  if (x.succ()) {
   x.succ(false);
   y.pred(x);
  }
  else y.left = x.right;

  x.right = y;
```

```
    x.balance(0);
    y.balance(0);
   }
   else
  {
    assert x.balance() == 1;

    w = x.right;
    x.right = w.left;
    w.left = x;
    y.left = w.right;
    w.right = y;
    if (w.balance() == -1) {
     x.balance(0);
     y.balance(1);
    }
    else if (w.balance() == 0) {
     x.balance(0);
     y.balance(0);
    }
    else {
     x.balance(-1);
     y.balance(0);
    }
    w.balance(0);


    if (w.pred()) {
     x.succ(w);
     w.pred(false);
    }
    if (w.succ()) {
     y.pred(w);
     w.succ(false);
    }

   }
  }
  else if (y.balance() == +2) {
   Entry KEY_GENERIC x = y.right;

   if (x.balance() == 1) {
    w = x;
    if (x.pred()) {
     x.pred(false);
     y.succ(x);
    }
    else y.right = x.left;
```

```
    x.left = y;
    x.balance(0);
    y.balance(0);
   }
   else {
    assert x.balance() == -1;

    w = x.left;
    x.left = w.right;
    w.right = x;
    y.right = w.left;
    w.left = y;
    if (w.balance() == 1) {
     x.balance(0);
     y.balance(-1);
    }
    else
if (w.balance() == 0) {
     x.balance(0);
     y.balance(0);
    }
    else {
     x.balance(1);
     y.balance(0);
    }
    w.balance(0);


    if (w.pred()) {
     y.succ(w);
     w.pred(false);
    }
    if (w.succ()) {
     x.pred(w);
     w.succ(false);
    }

   }
  }
  else return true;

  if (z == null) tree = w;
  else {
   if (z.left == y) z.left = w;
   else z.right = w;
  }
 }
```

```
 return true;
}




/** Finds the parent of an entry.
 *
 * @param e a node of the tree.
 * @return the parent of the given node, or {@code null} for the root.
 */

private Entry KEY_GENERIC parent(final Entry KEY_GENERIC e) {
 if (e == tree) return null;

 Entry KEY_GENERIC x, y, p;
 x = y = e;

 while(true) {
  if (y.succ()) {
   p = y.right;
   if (p == null || p.left != e) {
    while(! x.pred()) x = x.left;
    p = x.left;
   }
   return p;
  }
  else if (x.pred()) {
   p = x.left;
   if (p == null || p.right != e)
{
    while(! y.succ()) y = y.right;
    p = y.right;
   }
   return p;
  }

  x = x.left;
  y = y.right;
 }
}



SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public boolean remove(final KEY_TYPE k) {
 if (tree == null) return false;
```

```
int cmp;
Entry KEY_GENERIC p = tree, q = null;
boolean dir = false;
final KEY_GENERIC_TYPE kk = KEY_GENERIC_CAST k;

while(true) {
 if ((cmp = compare(kk, p.key)) == 0) break;
 else if (dir = cmp > 0) {
  q = p;
  if ((p = p.right()) == null) return false;
 }
 else {
  q = p;
  if ((p = p.left()) == null) return false;
 }
}

if (p.left == null) firstEntry = p.next();
if (p.right == null) lastEntry = p.prev();

if (p.succ()) {
 if (p.pred()) {
  if (q != null) {
   if (dir) q.succ(p.right);
   else q.pred(p.left);
  }
  else tree = dir ? p.right : p.left;
 }
 else {
  p.prev().right = p.right;

  if (q != null) {
   if (dir) q.right = p.left;
   else q.left = p.left;
  }
  else tree =
p.left;
 }
}
else {
 Entry KEY_GENERIC r = p.right;

 if (r.pred()) {
  r.left = p.left;
  r.pred(p.pred());
  if (! r.pred()) r.prev().right = r;
  if (q != null) {
```

```
  if (dir) q.right = r;
  else q.left = r;
 }
 else tree = r;

 r.balance(p.balance());
 q = r;
 dir = true;

}
else {
 Entry KEY_GENERIC s;

 while(true) {
  s = r.left;
  if (s.pred()) break;
  r = s;
 }

 if (s.succ()) r.pred(s);
 else r.left = s.right;

 s.left = p.left;

 if (! p.pred()) {
  p.prev().right = s;
  s.pred(false);
 }

 s.right = p.right;
 s.succ(false);

 if (q != null) {
  if (dir) q.right = s;
  else q.left = s;
 }
 else tree = s;

 s.balance(p.balance());
 q = r;
 dir = false;
}
}

Entry KEY_GENERIC y;

while(q != null) {
 y = q;
```

```
  q = parent(y);

  if (! dir) {
   dir = q != null && q.left != y;
   y.incBalance();

   if (y.balance() == 1) break;
   else if (y.balance()
== 2) {

    Entry KEY_GENERIC x = y.right;
    assert x != null;

    if (x.balance() == -1) {
     Entry KEY_GENERIC w;

     assert x.balance() == -1;

     w = x.left;
     x.left = w.right;
     w.right = x;
     y.right = w.left;
     w.left = y;

     if (w.balance() == 1) {
      x.balance(0);
      y.balance(-1);
     }
     else if (w.balance() == 0) {
      x.balance(0);
      y.balance(0);
     }
     else {
      assert w.balance() == -1;

      x.balance(1);
      y.balance(0);
     }

     w.balance(0);

     if (w.pred()) {
      y.succ(w);
      w.pred(false);
     }
     if (w.succ()) {
      x.pred(w);
      w.succ(false);
```

```
      }

      if (q != null) {
       if (dir) q.right = w;
       else q.left = w;
      }
      else tree = w;
     }
     else {
      if (q != null) {
       if (dir) q.right = x;
       else q.left = x;
      }
      else tree = x;

      if (x.balance() == 0) {
       y.right = x.left;
       x.left = y;
       x.balance(-1);
       y.balance(+1);
       break;
      }

      assert
x.balance() == 1;

      if (x.pred()) {
       y.succ(true);
       x.pred(false);
      }
      else y.right = x.left;

      x.left = y;
      y.balance(0);
      x.balance(0);
     }
    }
   }
   else {
    dir = q != null && q.left != y;
    y.decBalance();

    if (y.balance() == -1) break;
    else if (y.balance() == -2) {

     Entry KEY_GENERIC x = y.left;
     assert x != null;
```

```
    if (x.balance() == 1) {
     Entry KEY_GENERIC w;

     assert x.balance() == 1;

     w = x.right;
     x.right = w.left;
     w.left = x;
     y.left = w.right;
     w.right = y;

     if (w.balance() == -1) {
      x.balance(0);
      y.balance(1);
     }
     else if (w.balance() == 0) {
      x.balance(0);
      y.balance(0);
     }
     else {
      assert w.balance() == 1;

      x.balance(-1);
      y.balance(0);
     }

     w.balance(0);

     if (w.pred()) {
      x.succ(w);
      w.pred(false);
     }
     if
(w.succ()) {
      y.pred(w);
      w.succ(false);
     }

     if (q != null) {
      if (dir) q.right = w;
      else q.left = w;
     }
     else tree = w;
    }
    else {
     if (q != null) {
      if (dir) q.right = x;
      else q.left = x;
```

```
      }
      else tree = x;

      if (x.balance() == 0) {
       y.left = x.right;
       x.right = y;
       x.balance(+1);
       y.balance(-1);
       break;
      }

      assert x.balance() == -1;

      if (x.succ()) {
       y.pred(true);
       x.succ(false);
      }
      else y.left = x.right;

      x.right = y;
      y.balance(0);
      x.balance(0);
     }
    }
   }
  }

  count--;
  return true;
 }

 SUPPRESS_WARNINGS_KEY_UNCHECKED
 @Override
 public boolean contains(final KEY_TYPE k) {
  return findKey(KEY_GENERIC_CAST k) != null;
 }

#if KEY_CLASS_Object
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public K get(final KEY_TYPE k) {
  final Entry KEY_GENERIC entry = findKey(KEY_GENERIC_CAST k);
  return
 entry == null ? null : entry.key;
 }
#endif

 @Override
 public void clear() {
```

```
 count = 0;
 tree = null;
 firstEntry = lastEntry = null;
}


/** This class represent an entry in a tree set.
 *
 * <p>We use the only "metadata", i.e., {@link Entry#info}, to store
 * information about balance, predecessor status and successor status.
 *
 * <p>Note that since the class is recursive, it can be
 * considered equivalently a tree.
 */

private static final class Entry KEY_GENERIC implements Cloneable {
 /** If the bit in this mask is true, {@link #right} points to a successor. */
 private static final int SUCC_MASK = 1 << 31;
 /** If the bit in this mask is true, {@link #left} points to a predecessor. */
 private static final int PRED_MASK = 1 << 30;
 /** The bits in this mask hold the node balance info. You can get it just by casting to byte. */
 private static final int BALANCE_MASK = 0xFF;
 /** The key of this entry. */
 KEY_GENERIC_TYPE key;
 /** The pointers to
the left and right subtrees. */
 Entry KEY_GENERIC left, right;
 /** This integers holds different information in different bits (see {@link #SUCC_MASK}, {@link
#PRED_MASK} and {@link #BALANCE_MASK}). */
 int info;

 Entry() {}

 /** Creates a new entry with the given key.
  *
  * @param k a key.
  */
 Entry(final KEY_GENERIC_TYPE k) {
  this.key = k;
  info = SUCC_MASK | PRED_MASK;
 }

 /** Returns the left subtree.
  *
  * @return the left subtree ({@code null} if the left
  * subtree is empty).
  */
 Entry KEY_GENERIC left() {
```

```
 return (info & PRED_MASK) != 0 ? null : left;
}

/** Returns the right subtree.
 *
 * @return the right subtree ({@code null} if the right
 * subtree is empty).
 */
Entry KEY_GENERIC right() {
 return (info & SUCC_MASK) != 0 ? null : right;
}

/** Checks whether the left pointer is really a predecessor.
 * @return true if the left pointer is a predecessor.
 */
boolean pred() {
 return (info & PRED_MASK) != 0;
}

/**
Checks whether the right pointer is really a successor.
 * @return true if the right pointer is a successor.
 */
boolean succ() {
 return (info & SUCC_MASK) != 0;
}

/** Sets whether the left pointer is really a predecessor.
 * @param pred if true then the left pointer will be considered a predecessor.
 */
void pred(final boolean pred) {
 if (pred) info |= PRED_MASK;
 else info &= ~PRED_MASK;
}

/** Sets whether the right pointer is really a successor.
 * @param succ if true then the right pointer will be considered a successor.
 */
void succ(final boolean succ) {
 if (succ) info |= SUCC_MASK;
 else info &= ~SUCC_MASK;
}

/** Sets the left pointer to a predecessor.
 * @param pred the predecessr.
 */
void pred(final Entry KEY_GENERIC pred) {
 info |= PRED_MASK;
```

```
 left = pred;
}

/** Sets the right pointer to a successor.
 * @param succ the successor.
 */
void succ(final Entry KEY_GENERIC succ) {
 info |= SUCC_MASK;
 right
= succ;
}

/** Sets the left pointer to the given subtree.
 * @param left the new left subtree.
 */
void left(final Entry KEY_GENERIC left) {
 info &= ~PRED_MASK;
 this.left = left;
}

/** Sets the right pointer to the given subtree.
 * @param right the new right subtree.
 */
void right(final Entry KEY_GENERIC right) {
 info &= ~SUCC_MASK;
 this.right = right;
}

/** Returns the current level of the node.
 * @return the current level of this node.
 */
int balance() {
 return (byte)info;
}

/** Sets the level of this node.
 * @param level the new level of this node.
 */
void balance(int level) {
 info &= ~BALANCE_MASK;
 info |= (level & BALANCE_MASK);
}

/** Increments the level of this node. */
void incBalance() {
 info = info & ~BALANCE_MASK | ((byte)info + 1) & 0xFF;
}
```

```
/** Decrements the level of this node. */
protected void decBalance() {
 info = info & ~BALANCE_MASK | ((byte)info - 1) & 0xFF;
}

/**
Computes the next entry in the set order.
 *
 * @return the next entry ({@code null}) if this is the last entry).
 */

Entry KEY_GENERIC next() {
 Entry KEY_GENERIC next = this.right;
 if ((info & SUCC_MASK) == 0) while ((next.info & PRED_MASK) == 0) next = next.left;
 return next;
}

/** Computes the previous entry in the set order.
 *
 * @return the previous entry ({@code null}) if this is the first entry).
 */

Entry KEY_GENERIC prev() {
 Entry KEY_GENERIC prev = this.left;
 if ((info & PRED_MASK) == 0) while ((prev.info & SUCC_MASK) == 0) prev = prev.right;
 return prev;
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public Entry KEY_GENERIC clone() {
 Entry KEY_GENERIC c;
 try {
  c = (Entry KEY_GENERIC)super.clone();
 }
 catch(CloneNotSupportedException cantHappen) {
  throw new InternalError();
 }

 c.key = key;
 c.info = info;

 return c;
}

public boolean equals(final Object o) {
 if (!(o instanceof Entry)) return
```

```
false;
  Entry KEY_GENERIC_WILDCARD e = (Entry KEY_GENERIC_WILDCARD)o;

  return KEY_EQUALS(key, e.key);
 }

 public int hashCode() {
  return KEY2JAVAHASH_NOT_NULL(key);
 }


 public String toString() {
  return String.valueOf(key);
 }

 /*
   public void prettyPrint() {
   prettyPrint(0);
   }

   public void prettyPrint(int level) {
   if (pred()) {
   for (int i = 0; i < level; i++)
   System.err.print("  ");
   System.err.println("pred: " + left);
   }
   else if (left != null)
   left.prettyPrint(level +1);
   for (int i = 0; i < level; i++)
   System.err.print("  ");
   System.err.println(key + " (" + level() + ")");
   if (succ()) {
   for (int i = 0; i < level; i++)
   System.err.print("  ");
   System.err.println("succ: " + right);
   }
   else if (right != null)
   right.prettyPrint(level + 1);
   }
 */
}

/*
  public void prettyPrint() {
  System.err.println("size: " + count);
  if (tree != null) tree.prettyPrint();
  }
*/
```

```
@Override
public
int size() {
 return count;
}

@Override
public boolean isEmpty() {
 return count == 0;
}

@Override
public KEY_GENERIC_TYPE FIRST() {
 if (tree == null) throw new NoSuchElementException();
 return firstEntry.key;
}

@Override
public KEY_GENERIC_TYPE LAST() {
 if (tree == null) throw new NoSuchElementException();
 return lastEntry.key;
}


/** An iterator on the whole range.
 *
 * <p>This class can iterate in both directions on a threaded tree.
 */

private class SetIterator implements KEY_LIST_ITERATOR KEY_GENERIC {
  /** The entry that will be returned by the next call to {@link java.util.ListIterator#previous()} (or {@code null} if
no previous entry exists). */
  Entry KEY_GENERIC prev;
  /** The entry that will be returned by the next call to {@link java.util.ListIterator#next()} (or {@code null} if no
next entry exists). */
  Entry KEY_GENERIC next;
  /** The last entry that was returned (or {@code null} if we did not iterate or used {@link #remove()}).
 */
  Entry KEY_GENERIC curr;
  /** The current index (in the sense of a {@link java.util.ListIterator}). Note that this value is not meaningful when
this {@link SetIterator} has been created using the nonempty constructor.*/
  int index = 0;

  SetIterator() {
   next = firstEntry;
  }
```

```
SetIterator(final KEY_GENERIC_TYPE k) {
 if ((next = locateKey(k)) != null) {
  if (compare(next.key, k) <= 0) {
   prev = next;
   next = next.next();
  }
  else prev = next.prev();
 }
}

@Override
public boolean hasNext() { return next != null; }
@Override
public boolean hasPrevious() { return prev != null; }

void updateNext() { next = next.next(); }

Entry KEY_GENERIC nextEntry() {
 if (! hasNext()) throw new NoSuchElementException();
 curr = prev = next;
 index++;
 updateNext();
 return curr;
}

@Override
public KEY_GENERIC_TYPE NEXT_KEY() { return nextEntry().key; }
@Override
public KEY_GENERIC_TYPE PREV_KEY() { return previousEntry().key; }

void
updatePrevious() { prev = prev.prev(); }

Entry KEY_GENERIC previousEntry() {
 if (! hasPrevious()) throw new NoSuchElementException();
 curr = next = prev;
 index--;
 updatePrevious();
 return curr;
}

@Override
public int nextIndex() { return index; }

@Override
public int previousIndex() { return index - 1; }

@Override
```

```
 public void remove() {
  if (curr == null) throw new IllegalStateException();
  /* If the last operation was a next(), we are removing an entry that preceeds
     the current index, and thus we must decrement it. */
  if (curr == prev) index--;
  next = prev = curr;
  updatePrevious();
  updateNext();
  AVL_TREE_SET.this.remove(curr.key);
  curr = null;
 }
}

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new SetIterator(); }

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
SetIterator(from); }

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator()
{ return actualComparator; }

@Override
public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) { return new
Subset(KEY_NULL, true, to, false); }

@Override
public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) { return new Subset(from,
false, KEY_NULL, true); }

@Override
public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_TYPE from, final KEY_GENERIC_TYPE
to) { return new Subset(from, false, to, false); }

/** A subset with given range.
 *
 * <p>This class represents a subset. One has to specify the left/right
 * limits (which can be set to -&infin; or &infin;). Since the subset is a
 * view on the set, at a given moment it could happen that the limits of
 * the range are not any longer in the main set. Thus, things such as
 * {@link java.util.SortedSet#first()} or {@link java.util.SortedSet#size()} must be always computed
 * on-the-fly.
 */
private final class Subset extends ABSTRACT_SORTED_SET KEY_GENERIC implements java.io.Serializable,
SORTED_SET KEY_GENERIC {
 private static final long serialVersionUID = -7046029254386353129L;
```

```
  /** The start of the subset range, unless {@link #bottom} is true. */
  KEY_GENERIC_TYPE from;
  /** The end of the subset range, unless {@link #top} is true. */
  KEY_GENERIC_TYPE to;
  /** If true, the subset range starts from -&infin;. */
  boolean bottom;
  /** If true, the subset range goes to &infin;. */
  boolean top;

  /** Creates a new subset with given key range.
   *
   * @param from the start of the subset range.
   * @param bottom if true, the first parameter is ignored and the range starts from -&infin;.
   * @param to the end of the subset range.
   * @param top if true, the third parameter is ignored and the range goes to &infin;.
   */
  public Subset(final KEY_GENERIC_TYPE from, final boolean bottom, final KEY_GENERIC_TYPE to, final
boolean top) {
   if (! bottom && ! top && AVL_TREE_SET.this.compare(from, to) > 0) throw new
IllegalArgumentException("Start element ("
 + from  + ") is larger than end element (" + to + ")");

   this.from = from;
   this.bottom = bottom;
   this.to = to;
   this.top = top;
  }

  @Override
  public void clear() {
   final SubsetIterator i = new SubsetIterator();
   while(i.hasNext()) {
    i.NEXT_KEY();
    i.remove();
   }
  }

  /** Checks whether a key is in the subset range.
   * @param k a key.
   * @return true if is the key is in the subset range.
   */
  final boolean in(final KEY_GENERIC_TYPE k) {
   return (bottom || AVL_TREE_SET.this.compare(k, from) >= 0) &&
    (top || AVL_TREE_SET.this.compare(k, to) < 0);
  }

  SUPPRESS_WARNINGS_KEY_UNCHECKED
```

```
  @Override
  public boolean contains(final KEY_TYPE k) {
   return in(KEY_GENERIC_CAST k) && AVL_TREE_SET.this.contains(k);
  }

  @Override
  public boolean add(final KEY_GENERIC_TYPE  k) {
   if (! in(k)) throw new IllegalArgumentException("Element (" + k + ") out of range [" + (bottom ? "-" :
String.valueOf(from)) + ", " + (top ? "-" : String.valueOf(to)) +
 ")");
   return AVL_TREE_SET.this.add(k);
  }

  SUPPRESS_WARNINGS_KEY_UNCHECKED
  @Override
  public boolean remove(final KEY_TYPE k) {
   if (! in(KEY_GENERIC_CAST k)) return false;
   return AVL_TREE_SET.this.remove(k);
  }

  @Override
  public int size() {
   final SubsetIterator i = new SubsetIterator();
   int n = 0;

   while(i.hasNext()) {
    n++;
    i.NEXT_KEY();
   }

   return n;
  }

  @Override
  public boolean isEmpty() { return ! new SubsetIterator().hasNext(); }

  @Override
  public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return actualComparator; }

  @Override
  public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new SubsetIterator(); }

  @Override
  public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
SubsetIterator(from); }

  @Override
  public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) {
```

```
 if (top) return new Subset(from, bottom, to, false);
 return compare(to, this.to) < 0 ? new Subset(from, bottom,
to, false) : this;
 }

 @Override
 public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) {
  if (bottom) return new Subset(from, false, to, top);
  return compare(from, this.from) > 0 ? new Subset(from, false, to, top) : this;
 }

 @Override
 public SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE to) {
  if (top && bottom) return new Subset(from, false, to, false);
  if (! top) to = compare(to, this.to) < 0 ? to : this.to;
  if (! bottom) from = compare(from, this.from) > 0 ? from : this.from;
     if (! top && ! bottom && from == this.from && to == this.to) return this;
  return new Subset(from, false, to, false);
 }

 /** Locates the first entry.
  *
  * @return the first entry of this subset, or {@code null} if the subset is empty.
  */
 public AVL_TREE_SET.Entry KEY_GENERIC firstEntry() {
  if (tree == null) return null;
  // If this subset goes to -infinity, we return the main set first entry; otherwise, we locate the start
of the set.
  AVL_TREE_SET.Entry KEY_GENERIC e;
  if (bottom) e = firstEntry;
  else {
   e = locateKey(from);
   // If we find either the start or something greater we're OK.
   if (compare(e.key, from) < 0) e = e.next();
  }
  // Finally, if this subset doesn't go to infinity, we check that the resulting key isn't greater than the end.
  if (e == null || ! top && compare(e.key, to) >= 0) return null;
  return e;
 }

 /** Locates the last entry.
  *
  * @return the last entry of this subset, or {@code null} if the subset is empty.
  */
 public AVL_TREE_SET.Entry KEY_GENERIC lastEntry() {
  if (tree == null) return null;
  // If this subset goes to infinity, we return the main set last entry; otherwise, we locate the end of the set.
  AVL_TREE_SET.Entry KEY_GENERIC e;
```

```
  if (top) e = lastEntry;
  else {
   e = locateKey(to);
   // If we find something smaller than the end we're OK.
   if (compare(e.key, to) >= 0) e = e.prev();
  }
  // Finally, if this subset
doesn't go to -infinity, we check that the resulting key isn't smaller than the start.
  if (e == null || ! bottom && compare(e.key, from) < 0) return null;
  return e;
  }

  @Override
  public KEY_GENERIC_TYPE FIRST() {
   AVL_TREE_SET.Entry KEY_GENERIC e = firstEntry();
   if (e == null) throw new NoSuchElementException();
   return e.key;
  }

  @Override
  public KEY_GENERIC_TYPE LAST() {
   AVL_TREE_SET.Entry KEY_GENERIC e = lastEntry();
   if (e == null) throw new NoSuchElementException();
   return e.key;
  }

  /** An iterator for subranges.
   *
   * <p>This class inherits from {@link SetIterator}, but overrides the methods that
   * update the pointer after a {@link java.util.ListIterator#next()} or {@link java.util.ListIterator#previous()}. If we
would
   * move out of the range of the subset we just overwrite the next or previous
   * entry with {@code null}.
   */
  private final class SubsetIterator extends SetIterator {
   SubsetIterator() {
    next = firstEntry();
   }

   SubsetIterator(final
KEY_GENERIC_TYPE k) {
    this();

    if (next != null) {
     if (! bottom && compare(k, next.key) < 0) prev = null;
     else if (! top && compare(k, (prev = lastEntry()).key) >= 0) next = null;
     else {
      next = locateKey(k);
```

```
    if (compare(next.key, k) <= 0) {
     prev = next;
     next = next.next();
    }
    else prev = next.prev();
   }
  }
 }

 @Override
 void updatePrevious() {
  prev = prev.prev();
  if (! bottom && prev != null && AVL_TREE_SET.this.compare(prev.key, from) < 0) prev = null;
 }

 @Override
 void updateNext() {
  next = next.next();
  if (! top && next != null && AVL_TREE_SET.this.compare(next.key, to) >= 0) next = null;
 }
}
}

/** Returns a deep copy of this tree set.
 *
 * <p>This method performs a deep copy of this tree set; the data stored in the
 * set, however, is not cloned. Note that this makes a difference only for object keys.
 *
 * @return a deep copy of this tree
set.
 */

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public Object clone() {
 AVL_TREE_SET KEY_GENERIC c;
 try {
  c = (AVL_TREE_SET KEY_GENERIC)super.clone();
 }
 catch(CloneNotSupportedException cantHappen) {
  throw new InternalError();
 }

 c.allocatePaths();

 if (count != 0) {
  // Also this apparently unfathomable code is derived from GNU libavl.
```

```
  Entry KEY_GENERIC e, p, q, rp = new Entry KEY_GENERIC_DIAMOND(), rq = new Entry
KEY_GENERIC_DIAMOND();

  p = rp;
  rp.left(tree);

  q = rq;
  rq.pred(null);

  while(true) {
   if (! p.pred()) {
    e = p.left.clone();
    e.pred(q.left);
    e.succ(q);
    q.left(e);

    p = p.left;
    q = q.left;
   }
   else {
    while(p.succ()) {
     p = p.right;

     if (p == null) {
      q.right = null;
      c.tree = rq.left;

      c.firstEntry = c.tree;
      while(c.firstEntry.left != null) c.firstEntry = c.firstEntry.left;
      c.lastEntry = c.tree;
      while(c.lastEntry.right != null) c.lastEntry
 = c.lastEntry.right;

      return c;
     }
     q = q.right;
    }

    p = p.right;
    q = q.right;
   }

   if (! p.succ()) {
    e = p.right.clone();
    e.succ(q.right);
    e.pred(q);
    q.right(e);
   }
```

```
  }
 }

 return c;
}


private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
 int n = count;
 SetIterator i = new SetIterator();

 s.defaultWriteObject();
 while(n-- != 0) s.WRITE_KEY(i.NEXT_KEY());
}


/** Reads the given number of entries from the input stream, returning the corresponding tree.
 *
 * @param s the input stream.
 * @param n the (positive) number of entries to read.
 * @param pred the entry containing the key that preceeds the first key in the tree.
 * @param succ the entry containing the key that follows the last key in the tree.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
private Entry KEY_GENERIC readTree(final java.io.ObjectInputStream s, final int n, final Entry KEY_GENERIC
pred, final Entry KEY_GENERIC
succ) throws java.io.IOException, ClassNotFoundException {
 if (n == 1) {
  final Entry KEY_GENERIC top = new Entry KEY_GENERIC_DIAMOND(KEY_GENERIC_CAST
s.READ_KEY());
  top.pred(pred);
  top.succ(succ);

  return top;
 }

 if (n == 2) {
  /* We handle separately this case so that recursion will
   *always* be on nonempty subtrees. */
  final Entry KEY_GENERIC top = new Entry KEY_GENERIC_DIAMOND(KEY_GENERIC_CAST
s.READ_KEY());
  top.right(new Entry KEY_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY()));
  top.right.pred(top);
  top.balance(1);
  top.pred(pred);
  top.right.succ(succ);

  return top;
```

```
 }

 // The right subtree is the largest one.
 final int rightN = n / 2, leftN = n - rightN - 1;

 final Entry KEY_GENERIC top = new Entry KEY_GENERIC_DIAMOND();

 top.left(readTree(s, leftN, pred, top));

 top.key = KEY_GENERIC_CAST s.READ_KEY();

 top.right(readTree(s, rightN, top, succ));

 if (n == (n & -n)) top.balance(1); // Quick test for determining whether
n is a power of 2.

 return top;
}


private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
 s.defaultReadObject();
 /* The storedComparator is now correctly set, but we must restore
    on-the-fly the actualComparator. */
 setActualComparator();
 allocatePaths();

 if (count != 0) {
  tree = readTree(s, count, null, null);
  Entry KEY_GENERIC e;

  e = tree;
  while(e.left() != null) e = e.left();
  firstEntry = e;

  e = tree;
  while(e.right() != null) e = e.right();
  lastEntry = e;
 }
}

#ifdef ASSERTS_CODE
 private static KEY_GENERIC int checkTree(Entry KEY_GENERIC e) {
  if (e == null) return 0;

  final int leftN = checkTree(e.left()), rightN = checkTree(e.right());
  if (leftN + e.balance() != rightN)
   throw new AssertionError("Mismatch between left tree size (" + leftN + "), right tree size (" + rightN + ") and
```

```
balance (" + e.balance() + ")");

  return Math.max(leftN , rightN) + 1;
 }
#endif

#ifdef
 TEST

 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#else
  return Integer.toBinaryString(r.nextInt());
#endif
 }

 private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
 private static java.text.FieldPosition p = new java.text.FieldPosition(0);

 private static String format(double d) {
  StringBuffer s = new StringBuffer();
  return format.format(d, s, p).toString();
 }

 private static void speedTest(int n, boolean comp) {
  int i, j;
  AVL_TREE_SET m;
  java.util.TreeSet t;
  KEY_TYPE k[] = new KEY_TYPE[n];
  KEY_TYPE nk[] = new KEY_TYPE[n];
  long ms;

  for(i = 0; i < n; i++) {
   k[i] = genKey();
   nk[i] = genKey();
  }

  double totAdd = 0, totYes = 0, totNo = 0, totIterFor = 0, totIterBack = 0,
 totRemYes = 0, d, dd;

  if (comp) {
   for(j = 0; j < 20; j++) {
```

```
t = new java.util.TreeSet();

/* We first add all pairs to t. */
for(i = 0; i < n;  i++) t.add(KEY2OBJ(k[i]));

/* Then we remove the first half and put it back. */
for(i = 0; i < n/2;  i++) t.remove(KEY2OBJ(k[i]));

ms = System.currentTimeMillis();
for(i = 0; i < n/2;  i++) t.add(KEY2OBJ(k[i]));
d = System.currentTimeMillis() - ms;

/* Then we remove the other half and put it back again. */
ms = System.currentTimeMillis();
for(i = n/2; i < n;  i++) t.remove(KEY2OBJ(k[i]));
dd = System.currentTimeMillis() - ms ;

ms = System.currentTimeMillis();
for(i = n/2; i < n;  i++) t.add(KEY2OBJ(k[i]));
d += System.currentTimeMillis() - ms;
if (j > 2) totAdd += n/d;
System.out.print("Add: " + format(n/d) +" K/s ");

/* Then we remove again the first half. */
ms = System.currentTimeMillis();
for(i = 0; i < n/2;  i++) t.remove(KEY2OBJ(k[i]));
dd +=
System.currentTimeMillis() - ms ;
if (j > 2) totRemYes += n/dd;
System.out.print("RemYes: " + format(n/dd) +" K/s ");

/* And then we put it back. */
for(i = 0; i < n/2;  i++) t.add(KEY2OBJ(k[i]));

/* We check for pairs in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n;  i++) t.contains(KEY2OBJ(k[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totYes += d;
System.out.print("Yes: " + format(d) +" K/s ");

/* We check for pairs not in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n;  i++) t.contains(KEY2OBJ(nk[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totNo += d;
System.out.print("No: " + format(d) +" K/s ");
```

```
    /* We iterate on t. */
    ms = System.currentTimeMillis();
    for(Iterator it = t.iterator(); it.hasNext(); it.next());
    d = 1.0 * n / (System.currentTimeMillis() - ms);
    if (j > 2) totIterFor += d;
    System.out.print("IterFor: " + format(d) +" K/s
  ");

    System.out.println();
   }

   System.out.println();
   System.out.println("java.util Add: " + format(totAdd/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s
Yes: " + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3)) + " K/s IterFor: " + format(totIterFor/(j-3))  + "
K/s");

   System.out.println();

   totAdd = totYes = totNo = totIterFor = totIterBack = totRemYes = 0;

  }

  for(j = 0; j < 20; j++) {

   m = new AVL_TREE_SET();

   /* We first add all pairs to m. */
   for(i = 0; i < n;  i++) m.add(k[i]);

   /* Then we remove the first half and put it back. */
   for(i = 0; i < n/2;  i++) m.remove(k[i]);

   ms = System.currentTimeMillis();
   for(i = 0; i < n/2;  i++) m.add(k[i]);
   d = System.currentTimeMillis() - ms;

   /* Then we remove the other half and put it back again. */
   ms = System.currentTimeMillis();
   for(i = n/2; i < n;  i++) m.remove(k[i]);
   dd = System.currentTimeMillis() - ms ;

   ms = System.currentTimeMillis();
   for(i = n/2;
  i < n;  i++) m.add(k[i]);
   d += System.currentTimeMillis() - ms;
   if (j > 2) totAdd += n/d;
   System.out.print("Add: " + format(n/d) +" K/s ");
```

```
  /* Then we remove again the first half. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n/2;  i++) m.remove(k[i]);
  dd += System.currentTimeMillis() - ms ;
  if (j > 2) totRemYes += n/dd;
  System.out.print("RemYes: " + format(n/dd) +" K/s ");

  /* And then we put it back. */
  for(i = 0; i < n/2;  i++) m.add(k[i]);

  /* We check for pairs in m. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) m.contains(k[i]);
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totYes += d;
  System.out.print("Yes: " + format(d) +" K/s ");

  /* We check for pairs not in m. */
  ms = System.currentTimeMillis();
  for(i = 0; i < n;  i++) m.contains(nk[i]);
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totNo += d;
  System.out.print("No: " + format(d) +" K/s ");

  /* We iterate
on m. */
  KEY_LIST_ITERATOR it = (KEY_LIST_ITERATOR)m.iterator();
  ms = System.currentTimeMillis();
  for(; it.hasNext(); it.NEXT_KEY());
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totIterFor += d;
  System.out.print("IterFor: " + format(d) +" K/s ");

  /* We iterate back on m. */
  ms = System.currentTimeMillis();
  for(; it.hasPrevious(); it.PREV_KEY());
  d = 1.0 * n / (System.currentTimeMillis() - ms);
  if (j > 2) totIterBack += d;
  System.out.print("IterBack: " + format(d) +" K/s ");

  System.out.println();
 }


 System.out.println();
 System.out.println("fastutil  Add: " + format(totAdd/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s
Yes: " + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3)) + " K/s IterFor: " + format(totIterFor/(j-3))  + " K/s
```

```
IterBack: " + format(totIterBack/(j-3)) + "K/s");

 System.out.println();
}


private static boolean valEquals(Object o1, Object o2) {
 return o1 == null ? o2 == null : o1.equals(o2);
}

private
static void fatal(String msg) {
 System.out.println(msg);
 System.exit(1);
}

private static void ensure(boolean cond, String msg) {
 if (cond) return;
 fatal(msg);
}

private static Object[] k, v, nk;
private static KEY_TYPE kt[];
private static KEY_TYPE nkt[];
private static AVL_TREE_SET topSet;

protected static void testSets(SORTED_SET m, SortedSet t, int n, int level) {
 long ms;
 boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement;
 boolean rt = false, rm = false;

 if (level > 4) return;


 /* Now we check that both sets agree on first/last keys. */

 mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

 try {
  m.first();
 }
 catch (NoSuchElementException e) { mThrowsNoElement = true; }
 try {
  t.first();
 }
 catch (NoSuchElementException e) { tThrowsNoElement = true; }

 ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): first() divergence at
```

```
 start in NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  if (! mThrowsNoElement) ensure(t.first().equals(m.first()), "Error (" + level + ", " + seed + "): m and t differ at start
on their first key (" + m.first() + ", " + t.first() +")");

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.last();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.last();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): last() divergence at start in
NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");


  if (! mThrowsNoElement) ensure(t.last().equals(m.last()), "Error (" + level + ", " + seed + "): m and t differ at start
on their last key (" + m.last() + ", " + t.last() +")");


  /* Now we check that m and t are equal. */
  if (!m.equals(t) || ! t.equals(m)) System.err.println("m:
 " + m + " t: " + t);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");



  /* Now we check that m actually holds that data. */
  for(Iterator i=t.iterator(); i.hasNext();) {
   ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */
  for(Iterator i=m.iterator(); i.hasNext();) {
   ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
m)");
  }

  /* Now we check that inquiries about random data give the same answer in m and t. For
     m we use the polymorphic method. */

  for(int i=0; i<n;  i++) {
```

```
  KEY_TYPE T = genKey();

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.contains(T);
  }
  catch
 (NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }

  try {
   t.contains(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  if (!mThrowsNoElement && !mThrowsIllegal) ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)),
"Error (" + level + ", " + seed + "): divergence in keys between t and m (polymorphic method)");
  }

  /* Again, we check that inquiries about random data give the same answer in m and t, but
    for m we
 use the standard method. */

  for(int i=0; i<n;  i++) {
  KEY_TYPE T = genKey();

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.contains(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }

  try {
   t.contains(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
```

```
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  if (!mThrowsNoElement && !mThrowsIllegal) ensure(m.contains(KEY2OBJ(T)) ==  t.contains(KEY2OBJ(T)),
 "Error (" + level + ", " + seed + "): divergence between t and m (standard method)");
  }

 /* Now we add and remove random data in m and t, checking that the result is the same. */

 for(int i=0; i<20*n;  i++) {
  KEY_TYPE T = genKey();

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   rm = m.add(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }

  try {
   rt = t.add(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }


  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() divergence in
IllegalArgumentException
 for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  if (!mThrowsNoElement && !mThrowsIllegal) ensure(rm == rt, "Error (" + level + ", " + seed + "): divergence in
add() between t and m");

  T = genKey();

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   rm = m.remove(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }

  try {
   rt = t.remove(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }
```

```
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }


  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
IllegalArgumentException
 for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  if (!mThrowsNoElement && !mThrowsIllegal) ensure(rm == rt, "Error (" + level + ", " + seed + "): divergence in
remove() between t and m");
  }

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal");

  /* Now we check that m actually holds the same data. */

  for(Iterator i=t.iterator(); i.hasNext();) {
   ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */

  for(Iterator i=m.iterator(); i.hasNext();) {
   ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
m)");
  }

  /* Now we check that both sets agree on first/last keys. */

  mThrowsNoElement = mThrowsIllegal
 = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.first();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.first();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): first() divergence in
NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  if (! mThrowsNoElement) ensure(t.first().equals(m.first()), "Error (" + level + ", " + seed + "): m and t differ on their
first key (" + m.first() + ", " + t.first() +")");
```

```
  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.last();
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.last();
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): last() divergence in
NoSuchElementException  (" + mThrowsNoElement
 + ", " + tThrowsNoElement + ")");

  if (! mThrowsNoElement) ensure(t.last().equals(m.last()), "Error (" + level + ", " + seed + "): m and t differ on their
last key (" + m.last() + ", " + t.last() +")");

  /* Now we check cloning. */

  if (level == 0) {
   ensure(m.equals(((AVL_TREE_SET)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
   ensure(((AVL_TREE_SET)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
  }

  int h = m.hashCode();


  /* Now we save and read m. */

  SORTED_SET m2 = null;

  try {
   java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
   java.io.OutputStream os = new java.io.FileOutputStream(ff);
   java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

   oos.writeObject(m);
   oos.close();

   java.io.InputStream is = new java.io.FileInputStream(ff);
   java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

   m2
 = (SORTED_SET)ois.readObject();
   ois.close();
   ff.delete();
  }
```

```
catch(Exception e) {
 e.printStackTrace();
 System.exit(1);
}

ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

/* Now we check that m2 actually holds that data. */

ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
/* Now we take out of m everything, and check that it is empty. */

for(Iterator i=t.iterator(); i.hasNext();) m2.remove(i.next());

ensure(m2.isEmpty(), "Error (" + level + ", " + seed + "): m2 is not empty (as it should be)");

/* Now we play with iterators. */

{
 java.util.ListIterator i, j;
 Object J;
 i = (java.util.ListIterator)m.iterator();
 j = new java.util.LinkedList(t).listIterator();

 for(int k = 0; k < 2*n; k++) {
  ensure(i.hasNext() == j.hasNext(),
"Error (" + level + ", " + seed + "): divergence in hasNext()");
  ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious()");

  if (r.nextFloat() < .8 && i.hasNext()) {
   ensure(i.next().equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next()");

   if (r.nextFloat() < 0.5) {
    i.remove();
    j.remove();
    t.remove(J);
   }
  }
  else if (r.nextFloat() < .2 && i.hasPrevious()) {
   ensure(i.previous().equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous()");

   if (r.nextFloat() < 0.5) {
    i.remove();
    j.remove();
    t.remove(J);
   }
  }
```

```
    ensure(i.nextIndex() == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
    ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

   }

  }

  {
   boolean badPrevious = false;
   Object previous
 = null;
   it.unimi.dsi.fastutil.BidirectionalIterator i;
   java.util.ListIterator j;
   Object I, J;
   KEY_TYPE from = genKey();
   j = new java.util.LinkedList(t).listIterator();
   while(j.hasNext()) {
    Object k = j.next();
    if (((Comparable)k).compareTo(KEY2OBJ(from)) > 0) {
     badPrevious = true;
     j.previous();
     break;
    }
    previous = k;
   }

   i = (it.unimi.dsi.fastutil.BidirectionalIterator)m.iterator(from);

   for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting
point " + from + ")");
    ensure(i.hasPrevious() == j.hasPrevious() || badPrevious && (i.hasPrevious() == (previous != null)), "Error (" +
level + ", " + seed + "): divergence in hasPrevious() (iterator with starting point " + from + ")");

    if (r.nextFloat() < .8 && i.hasNext()) {
     ensure((I = i.next()).equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next()
 (" + I + ", " + J + ", iterator with starting point " + from + ")");
     //System.err.println("Done next " + I + " " + J + "  " + badPrevious);

     badPrevious = false;

     if (r.nextFloat() < 0.5) {
      //System.err.println("Removing in next");
      i.remove();
      j.remove();
      t.remove(J);
     }
```

```
    }
    else if (!badPrevious && r.nextFloat() < .2 && i.hasPrevious()) {
     ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

     if (r.nextFloat() < 0.5) {
      //System.err.println("Removing in prev");
      i.remove();
      j.remove();
      t.remove(J);
     }
    }
   }

  }

  /* Now we check that m actually holds that data. */

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after iteration");

  /* Now we select
 a pair of keys and create a subset. */

  if (! m.isEmpty()) {
   java.util.ListIterator i;
   Object start = m.first(), end = m.first();
   for(i = (java.util.ListIterator)m.iterator(); i.hasNext() && r.nextFloat() < .3; start = end = i.next());
   for(; i.hasNext() && r.nextFloat() < .95; end = i.next());

   //System.err.println("Checking subSet from " + start + " to " + end + " (level=" + (level+1) + ")...");
   testSets((SORTED_SET)m.subSet((KEY_CLASS)start, (KEY_CLASS)end), t.subSet(start, end), n, level + 1);

   ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after subSet");
   ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subSet");

   //System.err.println("Checking headSet to " + end + " (level=" + (level+1) + ")...");
   testSets((SORTED_SET)m.headSet((KEY_CLASS)end), t.headSet(end), n, level + 1);

   ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after headSet");
   ensure(t.equals(m), "Error
(" + level + ", " + seed + "): ! t.equals(m) after headSet");

   //System.err.println("Checking tailSet from " + start + " (level=" + (level+1) + ")...");
   testSets((SORTED_SET)m.tailSet((KEY_CLASS)start), t.tailSet(start), n, level + 1);

   ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after tailSet");
   ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after tailSet");
```

```
  }


 }


 private static void runTest(int n) {
  AVL_TREE_SET m = new AVL_TREE_SET();
  SortedSet t = new java.util.TreeSet();
  topSet = m;
  k = new Object[n];
  nk = new Object[n];
  kt = new KEY_TYPE[n];
  nkt = new KEY_TYPE[n];

  for(int i = 0; i < n; i++) {
#if KEY_CLASS_Object
   k[i] = kt[i] = genKey();
   nk[i] = nkt[i] = genKey();
#else
   k[i] = new KEY_CLASS(kt[i] = genKey());
   nk[i] = new KEY_CLASS(nkt[i] = genKey());
#endif
  }

  /* We add pairs to t. */
  for(int i = 0; i < n;  i++) t.add(k[i]);

  /* We add to m the same data */
  m.addAll(t);

  testSets(m,
 t, n, 0);

  System.out.println("Test OK");
  return;
 }


 public static void main(String args[]) {
  int n  = Integer.parseInt(args[1]);
  if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));


  try {
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
   else if ("test".equals(args[0])) runTest(n);
  } catch(Throwable e) {
```

```
  e.printStackTrace(System.err);
  System.err.println("seed: " + seed);
 }
}

#endif


}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AVLTreeSet.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterable;
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;
import it.unimi.dsi.fastutil.objects.ObjectSortedSet;
import it.unimi.dsi.fastutil.objects.ObjectSortedSets;
import PACKAGE.SORTED_MAP.FastSortedEntrySet;

import java.util.Comparator;
import
 java.util.Map;
import java.util.SortedMap;
import java.util.NoSuchElementException;

/** A class providing static methods and objects that do useful things with type-specific sorted maps.
*
```

```
* @see java.util.Collections
*/

public final class SORTED_MAPS {

 private SORTED_MAPS() {}

 /** Returns a comparator for entries based on a given comparator on keys.
  *
  * @param comparator a comparator on keys.
  * @return the associated comparator on entries.
  */
 public static KEY_GENERIC Comparator<? super Map.Entry<KEY_GENERIC_CLASS, ?>>
entryComparator(final KEY_COMPARATOR KEY_SUPER_GENERIC comparator) {
  return (Comparator<Map.Entry<KEY_GENERIC_CLASS, ?>>) (x, y) ->
comparator.compare(KEY_CLASS2TYPE(x.getKey()), KEY_CLASS2TYPE(y.getKey()));
 }

 /** Returns a bidirectional iterator that will be {@linkplain FastSortedEntrySet fast}, if possible, on the {@linkplain
Map#entrySet() entry set} of the provided {@code map}.
  * @param map a map from which we will try
 to extract a (fast) bidirectional iterator on the entry set.
  * @return a bidirectional iterator on the entry set of the given map that will be fast, if possible.
  * @since 8.0.0
  */
 SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
 public static KEY_VALUE_GENERIC ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC>
fastIterator(SORTED_MAP KEY_VALUE_GENERIC map) {
  final ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> entries = map.ENTRYSET();
  return entries instanceof SORTED_MAP.FastSortedEntrySet ? ((SORTED_MAP.FastSortedEntrySet
KEY_VALUE_GENERIC) entries).fastIterator() : entries.iterator();
 }

 /** Returns an iterable yielding a bidirectional iterator that will be {@linkplain FastSortedEntrySet fast}, if possible,
on the {@linkplain Map#entrySet() entry set} of the provided {@code map}.
  * @param map a map from which we will try to extract an iterable yielding a (fast) bidirectional iterator on the
entry set.
  * @return an iterable yielding a bidirectional iterator on the entry set
 of the given map that will be fast, if possible.
  * @since 8.0.0
  */
 SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
 public static KEY_VALUE_GENERIC ObjectBidirectionalIterable<MAP.Entry KEY_VALUE_GENERIC>
fastIterable(SORTED_MAP KEY_VALUE_GENERIC map) {
  final ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> entries = map.ENTRYSET();
  return entries instanceof SORTED_MAP.FastSortedEntrySet ? ((SORTED_MAP.FastSortedEntrySet
KEY_VALUE_GENERIC)entries)::fastIterator : entries;
 }
```

```
 /** An immutable class representing an empty type-specific sorted map.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific sorted map.
  */

 public static class EmptySortedMap KEY_VALUE_GENERIC extends MAPS.EmptyMap
KEY_VALUE_GENERIC implements SORTED_MAP KEY_VALUE_GENERIC, java.io.Serializable, Cloneable
{

  private static final long serialVersionUID = -7046029254386353129L;

  protected EmptySortedMap() {}

  @Override
  public KEY_COMPARATOR KEY_SUPER_GENERIC comparator()
 { return null; }

  @SuppressWarnings("unchecked")
  @Override
  public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { return
ObjectSortedSets.EMPTY_SET; }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
#else
  /** {@inheritDoc} */
#endif
  @Override
  @SuppressWarnings("unchecked")
  public ObjectSortedSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
ObjectSortedSets.EMPTY_SET; }

  SUPPRESS_WARNINGS_KEY_UNCHECKED
  @Override
  public SORTED_SET KEY_GENERIC keySet() { return SORTED_SETS.EMPTY_SET; }

  SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_TYPE from, final
KEY_GENERIC_TYPE to) { return EMPTY_MAP; }

  SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
  @Override
```

```
  public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_TYPE to) { return
EMPTY_MAP; }

  SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
  @Override
  public
 SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_TYPE from) { return EMPTY_MAP; }

  @Override
  public KEY_GENERIC_TYPE FIRST_KEY() { throw new NoSuchElementException(); }

  @Override
  public KEY_GENERIC_TYPE LAST_KEY() { throw new NoSuchElementException(); }

#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC headMap(KEY_GENERIC_CLASS oto) { return
headMap(KEY_CLASS2TYPE(oto)); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC tailMap(KEY_GENERIC_CLASS ofrom) { return
tailMap(KEY_CLASS2TYPE(ofrom)); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_CLASS ofrom,
 KEY_GENERIC_CLASS oto) { return subMap(KEY_CLASS2TYPE(ofrom), KEY_CLASS2TYPE(oto)); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS firstKey() { return KEY2OBJ(FIRST_KEY()); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS lastKey() { return KEY2OBJ(LAST_KEY()); }
#endif
```

```
 }



 /** An empty sorted map (immutable). It is serializable and cloneable.
  */
 SUPPRESS_WARNINGS_KEY_VALUE_RAWTYPES
 public static final EmptySortedMap EMPTY_MAP = new EmptySortedMap();

#if KEYS_REFERENCE || VALUES_REFERENCE
 /** Returns an empty sorted map (immutable). It is serializable and cloneable.
  *
  * <p>This method provides a typesafe access to {@link #EMPTY_MAP}.
  * @return an empty sorted map (immutable).
  */
 @SuppressWarnings("unchecked")
 public static KEY_VALUE_GENERIC SORTED_MAP
 KEY_VALUE_GENERIC emptyMap() {
  return EMPTY_MAP;
 }
#endif

 /** An immutable class representing a type-specific singleton sorted map.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific sorted map.
  */

 public static class Singleton KEY_VALUE_GENERIC extends MAPS.Singleton KEY_VALUE_GENERIC
implements SORTED_MAP KEY_VALUE_GENERIC, java.io.Serializable, Cloneable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final KEY_COMPARATOR KEY_SUPER_GENERIC comparator;

  protected Singleton(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value,
KEY_COMPARATOR KEY_SUPER_GENERIC comparator) {
   super(key, value);
   this.comparator = comparator;
  }

  protected Singleton(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value) {
   this(key, value, null);
  }

  SUPPRESS_WARNINGS_KEY_UNCHECKED
  final int compare(final KEY_GENERIC_TYPE k1, final KEY_GENERIC_TYPE k2) {
   return comparator == null ?
```

```
 KEY_CMP(k1, k2) : comparator.compare(k1, k2);
  }

  @Override
  public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return comparator; }

  SUPPRESS_WARNINGS_KEY_UNCHECKED
  @Override
  public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { if (entries == null) entries =
ObjectSortedSets.singleton(new ABSTRACT_MAP.BasicEntry KEY_VALUE_GENERIC_DIAMOND(key,
value), entryComparator(comparator)); return (ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC>)entries; }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
#else
  /** {@inheritDoc} */
#endif
  @Override
  @SuppressWarnings({ "rawtypes", "unchecked" })
  public ObjectSortedSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
(ObjectSortedSet)ENTRYSET(); }

  @Override
  public SORTED_SET KEY_GENERIC keySet() { if (keys == null) keys = SORTED_SETS.singleton(key,
comparator); return (SORTED_SET KEY_GENERIC)keys;
  }

  SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_TYPE from, final
KEY_GENERIC_TYPE to) { if (compare(from, key) <= 0 && compare(key, to) < 0) return this; return
EMPTY_MAP; }

  SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_TYPE to) { if (compare(key,
to) < 0) return this; return EMPTY_MAP; }

  SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_TYPE from) { if (compare(from,
key) <= 0) return this; return EMPTY_MAP; }

  @Override
  public KEY_GENERIC_TYPE FIRST_KEY() { return key; }
```

```
  @Override
  public KEY_GENERIC_TYPE LAST_KEY() { return key; }

#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC headMap(KEY_GENERIC_CLASS oto) { return
headMap(KEY_CLASS2TYPE(oto));
 }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC tailMap(KEY_GENERIC_CLASS ofrom) { return
tailMap(KEY_CLASS2TYPE(ofrom)); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_CLASS ofrom,
KEY_GENERIC_CLASS oto) { return subMap(KEY_CLASS2TYPE(ofrom), KEY_CLASS2TYPE(oto)); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS firstKey() { return KEY2OBJ(FIRST_KEY()); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS lastKey() { return KEY2OBJ(LAST_KEY()); }
#endif
 }

 /** Returns a type-specific immutable sorted map containing
 only the specified pair. The returned sorted map is serializable and cloneable.
  *
  * <p>Note that albeit the returned map is immutable, its default return value may be changed.
  *
  * @param key the only key of the returned sorted map.
  * @param value the only value of the returned sorted map.
  * @return a type-specific immutable sorted map containing just the pair {@code &lt;key,value&gt;}.
```

```
 */

 public static KEY_VALUE_GENERIC SORTED_MAP KEY_VALUE_GENERIC singleton(final
KEY_GENERIC_CLASS key, VALUE_GENERIC_CLASS value) { return new Singleton
KEY_VALUE_GENERIC_DIAMOND(KEY_CLASS2TYPE(key), VALUE_CLASS2TYPE(value));}

 /** RETURNS a type-specific immutable sorted map containing only the specified pair. The returned sorted map is
serializable and cloneable.
  *
  * <p>Note that albeit the returned map is immutable, its default return value may be changed.
  *
  * @param key the only key of the returned sorted map.
  * @param value the only value of the returned sorted map.

 * @param comparator the comparator to use in the returned sorted map.
  * @return a type-specific immutable sorted map containing just the pair {@code &lt;key,value&gt;}.
  */

 public static KEY_VALUE_GENERIC SORTED_MAP KEY_VALUE_GENERIC singleton(final
KEY_GENERIC_CLASS key, VALUE_GENERIC_CLASS value, KEY_COMPARATOR
KEY_SUPER_GENERIC comparator) { return new Singleton
KEY_VALUE_GENERIC_DIAMOND(KEY_CLASS2TYPE(key), VALUE_CLASS2TYPE(value), comparator);
}

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

 /** Returns a type-specific immutable sorted map containing only the specified pair. The returned sorted map is
serializable and cloneable.
  *
  * <p>Note that albeit the returned map is immutable, its default return value may be changed.
  *
  * @param key the only key of the returned sorted map.
  * @param value the only value of the returned sorted map.
  * @return a type-specific immutable sorted map containing just the pair {@code &lt;key,value&gt;}.
  */

 public static KEY_VALUE_GENERIC
 SORTED_MAP KEY_VALUE_GENERIC singleton(final KEY_GENERIC_TYPE key, final
VALUE_GENERIC_TYPE value) {
  return new Singleton KEY_VALUE_GENERIC_DIAMOND(key, value);
 }

 /** Returns a type-specific immutable sorted map containing only the specified pair. The returned sorted map is
serializable and cloneable.
  *
  * <p>Note that albeit the returned map is immutable, its default return value may be changed.
  *
  * @param key the only key of the returned sorted map.
```

```
  * @param value the only value of the returned sorted map.
  * @param comparator the comparator to use in the returned sorted map.
  * @return a type-specific immutable sorted map containing just the pair {@code &lt;key,value&gt;}.
  */

 public static KEY_VALUE_GENERIC SORTED_MAP KEY_VALUE_GENERIC singleton(final
KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value, KEY_COMPARATOR
KEY_SUPER_GENERIC comparator) {
  return new Singleton KEY_VALUE_GENERIC_DIAMOND(key, value, comparator);
 }

#endif


  /** A synchronized wrapper
 class for sorted maps. */

 public static class SynchronizedSortedMap KEY_VALUE_GENERIC extends MAPS.SynchronizedMap
KEY_VALUE_GENERIC implements SORTED_MAP KEY_VALUE_GENERIC, java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final SORTED_MAP KEY_VALUE_GENERIC sortedMap;

  protected SynchronizedSortedMap(final SORTED_MAP KEY_VALUE_GENERIC m, final Object sync) {
   super(m, sync);
   sortedMap = m;
  }

  protected SynchronizedSortedMap(final SORTED_MAP KEY_VALUE_GENERIC m) {
   super(m);
   sortedMap = m;
  }

  @Override
  public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { synchronized(sync) { return
sortedMap.comparator(); } }

  @Override
  public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { if (entries == null) entries =
ObjectSortedSets.synchronize(sortedMap.ENTRYSET(), sync); return (ObjectSortedSet<MAP.Entry
KEY_VALUE_GENERIC>)entries; }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
  /** {@inheritDoc}

 * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
```

```
#else
  /** {@inheritDoc} */
#endif
  @Override
  @SuppressWarnings({ "rawtypes", "unchecked" })
  public ObjectSortedSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
(ObjectSortedSet)ENTRYSET(); }

  @Override
  public SORTED_SET KEY_GENERIC keySet() { if (keys == null) keys =
SORTED_SETS.synchronize(sortedMap.keySet(), sync); return (SORTED_SET KEY_GENERIC)keys; }

  @Override
  public SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_TYPE from, final
KEY_GENERIC_TYPE to) { return new SynchronizedSortedMap
KEY_VALUE_GENERIC_DIAMOND(sortedMap.subMap(from, to), sync); }

  @Override
  public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_TYPE to) { return new
SynchronizedSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.headMap(to), sync); }

  @Override
  public SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_TYPE from) { return new
SynchronizedSortedMap
 KEY_VALUE_GENERIC_DIAMOND(sortedMap.tailMap(from), sync); }

  @Override
  public KEY_GENERIC_TYPE FIRST_KEY() {  synchronized(sync) { return sortedMap.FIRST_KEY(); } }

  @Override
  public KEY_GENERIC_TYPE LAST_KEY() {  synchronized(sync) { return sortedMap.LAST_KEY(); } }

#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS firstKey() {  synchronized(sync) { return sortedMap.firstKey(); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS lastKey() {  synchronized(sync) { return sortedMap.lastKey(); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
```

```
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_CLASS from, final
KEY_GENERIC_CLASS to) { return new SynchronizedSortedMap
 KEY_VALUE_GENERIC_DIAMOND(sortedMap.subMap(from, to), sync); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_CLASS to) { return new
SynchronizedSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.headMap(to), sync); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_CLASS from) { return new
SynchronizedSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.tailMap(from), sync); }
#endif


 }

 /** Returns a synchronized type-specific sorted map backed by the given type-specific sorted map.
  *
  * @param m the sorted map to be wrapped in a synchronized sorted map.
  * @return a synchronized view of the specified sorted map.
  * @see java.util.Collections#synchronizedSortedMap(SortedMap)
  */
 public static KEY_VALUE_GENERIC
 SORTED_MAP KEY_VALUE_GENERIC synchronize(final SORTED_MAP KEY_VALUE_GENERIC m) {
return new SynchronizedSortedMap KEY_VALUE_GENERIC_DIAMOND(m); }

 /** Returns a synchronized type-specific sorted map backed by the given type-specific sorted map, using an
assigned object to synchronize.
  *
  * @param m the sorted map to be wrapped in a synchronized sorted map.
  * @param sync an object that will be used to synchronize the access to the sorted sorted map.
  * @return a synchronized view of the specified sorted map.
  * @see java.util.Collections#synchronizedSortedMap(SortedMap)
  */

 public static KEY_VALUE_GENERIC SORTED_MAP KEY_VALUE_GENERIC synchronize(final
SORTED_MAP KEY_VALUE_GENERIC m, final Object sync) { return new SynchronizedSortedMap
KEY_VALUE_GENERIC_DIAMOND(m, sync); }
```

```
 /** An unmodifiable wrapper class for sorted maps. */

 public static class UnmodifiableSortedMap KEY_VALUE_GENERIC extends MAPS.UnmodifiableMap
KEY_VALUE_GENERIC implements SORTED_MAP KEY_VALUE_GENERIC,
 java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final SORTED_MAP KEY_VALUE_GENERIC sortedMap;

  protected UnmodifiableSortedMap(final SORTED_MAP KEY_VALUE_GENERIC m) {
   super(m);
   sortedMap = m;
  }

  @Override
  public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return sortedMap.comparator(); }

  @Override
  public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { if (entries == null) entries =
ObjectSortedSets.unmodifiable(sortedMap.ENTRYSET()); return (ObjectSortedSet<MAP.Entry
KEY_VALUE_GENERIC>)entries; }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
#else
  /** {@inheritDoc} */
#endif
  @Override
  @SuppressWarnings({ "rawtypes", "unchecked" })
  public ObjectSortedSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
(ObjectSortedSet)ENTRYSET(); }

  @Override
  public
 SORTED_SET KEY_GENERIC keySet() { if (keys == null) keys =
SORTED_SETS.unmodifiable(sortedMap.keySet()); return (SORTED_SET KEY_GENERIC)keys; }

  @Override
  public SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_TYPE from, final
KEY_GENERIC_TYPE to) { return new UnmodifiableSortedMap
KEY_VALUE_GENERIC_DIAMOND(sortedMap.subMap(from, to)); }

  @Override
  public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_TYPE to) { return new
```

```
UnmodifiableSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.headMap(to)); }

  @Override
  public SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_TYPE from) { return new
UnmodifiableSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.tailMap(from)); }

  @Override
  public KEY_GENERIC_TYPE FIRST_KEY() {  return sortedMap.FIRST_KEY(); }

  @Override
  public KEY_GENERIC_TYPE LAST_KEY() {  return sortedMap.LAST_KEY(); }

#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public
 KEY_GENERIC_CLASS firstKey() {  return sortedMap.firstKey(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS lastKey() {  return sortedMap.lastKey(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_CLASS from, final
KEY_GENERIC_CLASS to) { return new UnmodifiableSortedMap
KEY_VALUE_GENERIC_DIAMOND(sortedMap.subMap(from, to)); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_CLASS to) { return new
UnmodifiableSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.headMap(to)); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public
 SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_CLASS from) { return new
UnmodifiableSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.tailMap(from)); }
```

```
#endif


 }

 /** Returns an unmodifiable type-specific sorted map backed by the given type-specific sorted map.
  *
  * @param m the sorted map to be wrapped in an unmodifiable sorted map.
  * @return an unmodifiable view of the specified sorted map.
  * @see java.util.Collections#unmodifiableSortedMap(SortedMap)
  */
 public static KEY_VALUE_GENERIC SORTED_MAP KEY_VALUE_GENERIC unmodifiable(final
SORTED_MAP KEY_VALUE_GENERIC m) { return new UnmodifiableSortedMap
KEY_VALUE_GENERIC_DIAMOND(m); }



#if defined(TEST) && ! KEY_CLASS_Reference

 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return
 r.NEXT_KEY();
#else
  return Integer.toBinaryString(r.nextInt());
#endif
 }

 private static VALUE_TYPE genValue() {
#if VALUE_CLASS_Byte || VALUE_CLASS_Short || VALUE_CLASS_Character
  return (VALUE_TYPE)(r.nextInt());
#elif VALUES_PRIMITIVE
  return r.NEXT_VALUE();
#elif !VALUE_CLASS_Reference || KEY_CLASS_Reference
  return Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }


 private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
 private static java.text.FieldPosition p = new java.text.FieldPosition(0);
```

```
private static String format(double d) {
 StringBuffer s = new StringBuffer();
 return format.format(d, s, p).toString();
}

private static void speedTest(int n, boolean comp) {
 System.out.println("There are presently no speed tests for this class.");
}


private static boolean valEquals(Object o1, Object o2) {
 return o1 == null ? o2 == null : o1.equals(o2);
}

private static void
fatal(String msg) {
 System.out.println(msg);
 System.exit(1);
}

private static void ensure(boolean cond, String msg) {
 if (cond) return;
 fatal(msg);
}

private static Object[] k, v, nk;
private static KEY_TYPE kt[];
private static KEY_TYPE nkt[];
private static VALUE_TYPE vt[];
private static SORTED_MAP topMap;

protected static void testMaps(SORTED_MAP m, SortedMap t, int n, int level) {
 long ms;
 boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement, mThrowsUnsupp,
tThrowsUnsupp;
 Object rt = null, rm = null;

 if (level > 1) return;


 /* Now we check that both maps agree on first/last keys. */

 mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

 try {
  m.firstKey();
 }
```

```
  catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.firstKey();
  }
  catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level
 + ", " + seed + "): firstKey() divergence at start in java.util.NoSuchElementException  (" + mThrowsNoElement +
", " + tThrowsNoElement + ")");
  if (! mThrowsNoElement) ensure(t.firstKey().equals(m.firstKey()), "Error (" + level + ", " + seed + "): m and t differ
at start on their first key (" + m.firstKey() + ", " + t.firstKey() +")");

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.lastKey();
  }
  catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.lastKey();
  }
  catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): lastKey() divergence at start
in java.util.NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");


  if (! mThrowsNoElement) ensure(t.lastKey().equals(m.lastKey()), "Error (" + level + ", " + seed + "): m and t differ
at start on their last key (" + m.lastKey()
 + ", " + t.lastKey() +")");


  /* Now we check that m and t are equal. */
  if (!m.equals(t) || ! t.equals(m)) System.err.println("m: " + m + " t: " + t);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");



  /* Now we check that m actually holds that data. */
  for(java.util.Iterator i=t.entrySet().iterator(); i.hasNext();) {
   java.util.Map.Entry e = (java.util.Map.Entry)i.next();
   ensure(valEquals(e.getValue(), m.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after insertion (iterating on t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */
```

```
  for(java.util.Iterator i=m.entrySet().iterator(); i.hasNext();) {
   java.util.Map.Entry e = (java.util.Map.Entry)i.next();
   ensure(valEquals(e.getValue(), t.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
 ("+e+") after insertion (iterating on m)");
  }

  /* Now we check that m actually holds the same keys. */
  for(java.util.Iterator i=t.keySet().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(m.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key ("+o+") after insertion
(iterating on t)");
   ensure(m.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key ("+o+", in keySet()) after
insertion (iterating on t)");
  }

  /* Now we check that m actually holds the same keys, but iterating on m. */
  for(java.util.Iterator i=m.keySet().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(t.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key after insertion (iterating on m)");
   ensure(t.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (in keySet()) after insertion
(iterating on m)");
  }


  /* Now we check that m actually hold the same values. */
  for(java.util.Iterator
 i=t.values().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(m.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after insertion (iterating on
t)");
   ensure(m.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
insertion (iterating on t)");
  }

  /* Now we check that m actually hold the same values, but iterating on m. */
  for(java.util.Iterator i=m.values().iterator(); i.hasNext();) {
   Object o = i.next();
   ensure(t.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after insertion (iterating on
m)");
   ensure(t.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
insertion (iterating on m)");
  }

  /* Now we check that inquiries about random data give the same answer in m and t. For
     m we use the polymorphic method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();
```

```
  mThrowsNoElement
 = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.containsKey(KEY2OBJ(T));
  }
  catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }

  try {
   t.containsKey(KEY2OBJ(T));
  }
  catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): containsKey() divergence in
java.util.NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): containsKey() divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  if (!mThrowsNoElement && !mThrowsIllegal) {
   ensure(m.containsKey(KEY2OBJ(T)) == t.containsKey(KEY2OBJ(T)), "Error (" + level + ", " + seed
 + "): divergence in keys between t and m (polymorphic method)");

#if KEY_CLASS_Object && ! (VALUES_REFERENCE)
   if ((m.GET_VALUE(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
    t.get(KEY2OBJ(T)) != null &&
    ! VALUE2OBJ(m.GET_VALUE(T)).equals(t.get(KEY2OBJ(T))))
#else
    if ((m.get(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
     t.get(KEY2OBJ(T)) != null &&
     ! m.get(KEY2OBJ(T)).equals(t.get(KEY2OBJ(T))))
#endif
     {
      System.out.println("Error (" + level + ", " + seed + "): divergence between t and m (polymorphic method)");
      System.exit(1);
     }
  }
 }

 /* Again, we check that inquiries about random data give the same answer in m and t, but
    for m we use the standard method. */

 for(int i=0; i<n;  i++) {
  KEY_TYPE T = genKey();
```

```
  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal
 = false;

  try {
   m.get(KEY2OBJ(T));
  }
  catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }

  try {
   t.get(KEY2OBJ(T));
  }
  catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): get() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): get() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(m.get(KEY2OBJ(T)), t.get(KEY2OBJ(T))),
"Error (" + level + ", " + seed + "): divergence between t and m (standard method)");
  }

  /* Now we put and remove random data
 in m and t, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();
   VALUE_TYPE U = genValue();

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = mThrowsUnsupp =
tThrowsUnsupp = false;

  try {
   rm = m.put(KEY2OBJ(T), VALUE2OBJ(U));
  }
  catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }
  catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

  try {
   rt = t.put(KEY2OBJ(T), VALUE2OBJ(U));
  }
  catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }
  catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): put() divergence in
```

```
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
   ensure(mThrowsIllegal
 == tThrowsIllegal, "Error (" + level + ", " + seed + "): put() divergence in IllegalArgumentException for " + T + "
(" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
   //ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): put() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsUnsupp) ensure(valEquals(rm, rt), "Error (" + level +
", " + seed + "): divergence in put() between t and m (" + rt + ", " + rm + ")");

   T = genKey();

   mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = mThrowsUnsupp =
tThrowsUnsupp = false;

   try {
    rm = m.remove(KEY2OBJ(T));
   }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    rt = t.remove(KEY2OBJ(T));
   }
   catch
 (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }


   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
   //ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): remove() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsUnsupp) ensure(valEquals(rm, rt), "Error (" + level +
", " + seed + "): divergence in remove() between
 t and m (" + rt + ", " + rm + ")");
  }

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal");

  /* Now we check that m actually holds the same data. */

  for(java.util.Iterator i=t.entrySet().iterator(); i.hasNext();) {
   java.util.Map.Entry e = (java.util.Map.Entry)i.next();
```

```
  ensure(valEquals(e.getValue(), m.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after removal (iterating on t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */

  for(java.util.Iterator i=m.entrySet().iterator(); i.hasNext();) {
   java.util.Map.Entry e = (java.util.Map.Entry)i.next();
  ensure(valEquals(e.getValue(), t.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after removal (iterating on m)");
  }

  /* Now we check that m actually holds the same keys. */

  for(java.util.Iterator
 i=t.keySet().iterator(); i.hasNext();) {
  Object o = i.next();
  ensure(m.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key ("+o+") after removal (iterating
on t)");
  ensure(m.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key ("+o+", in keySet()) after
removal (iterating on t)");
  }

  /* Now we check that m actually holds the same keys, but iterating on m. */

  for(java.util.Iterator i=m.keySet().iterator(); i.hasNext();) {
  Object o = i.next();
  ensure(t.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key after removal (iterating on m)");
  ensure(t.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (in keySet()) after removal
(iterating on m)");
  }


  /* Now we check that m actually hold the same values. */

  for(java.util.Iterator i=t.values().iterator(); i.hasNext();) {
  Object o = i.next();
  ensure(m.containsValue(o),
 "Error (" + level + ", " + seed + "): m and t differ on a value after removal (iterating on t)");
  ensure(m.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
removal (iterating on t)");
  }

  /* Now we check that m actually hold the same values, but iterating on m. */

  for(java.util.Iterator i=m.values().iterator(); i.hasNext();) {
  Object o = i.next();
  ensure(t.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after removal (iterating on
```

```
m)");
   ensure(t.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after removal
(iterating on m)");
  }

  /* Now we check that both maps agree on first/last keys. */

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.firstKey();
  }
  catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.firstKey();
  }
  catch (java.util.NoSuchElementException
 e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): firstKey() divergence in
java.util.NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
  if (! mThrowsNoElement) ensure(t.firstKey().equals(m.firstKey()), "Error (" + level + ", " + seed + "): m and t differ
on their first key (" + m.firstKey() + ", " + t.firstKey() +")");

  mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

  try {
   m.lastKey();
  }
  catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
  try {
   t.lastKey();
  }
  catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): lastKey() divergence in
java.util.NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");

  if (! mThrowsNoElement) ensure(t.lastKey().equals(m.lastKey()), "Error
 (" + level + ", " + seed + "): m and t differ on their last key (" + m.lastKey() + ", " + t.lastKey() +")");

  /* Now we check cloning. */

  if (level == 0) {
   ensure(m.equals(((Singleton)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
   ensure(((Singleton)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
   m = (SORTED_MAP)((Singleton)m).clone();
  }
```

```
  int h = m.hashCode();


  /* Now we save and read m. */

  SORTED_MAP m2 = null;

  try {
   java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
   java.io.OutputStream os = new java.io.FileOutputStream(ff);
   java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

   oos.writeObject(m);
   oos.close();

   java.io.InputStream is = new java.io.FileInputStream(ff);
   java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

   m2 = (SORTED_MAP)ois.readObject();
   ois.close();
   ff.delete();
  }
  catch(Exception
 e) {
   e.printStackTrace();
   System.exit(1);
  }

#if !VALUE_CLASS_Reference
  ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

  /* Now we check that m2 actually holds that data. */

  ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
  ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
  /* Now we take out of m everything, and check that it is empty. */
#endif

  /* Now we play with iterators. */

  {
   java.util.ListIterator i, j;
   Object J;
   i = (java.util.ListIterator)m.entrySet().iterator();
   j = new java.util.LinkedList(t.entrySet()).listIterator();

   for(int k = 0; k < 2*n; k++) {
```

```
    ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext()");
    ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious()");

    if
 (r.nextFloat() < .8 && i.hasNext()) {
     ensure(((java.util.Map.Entry)i.next()).getKey().equals(J = ((Map.Entry)j.next()).getKey()), "Error (" + level + ", " +
seed + "): divergence in next()");

    }
    else if (r.nextFloat() < .2 && i.hasPrevious()) {
     ensure(((java.util.Map.Entry)i.previous()).getKey().equals(J = ((Map.Entry)j.previous()).getKey()), "Error (" + level
+ ", " + seed + "): divergence in previous()");

    }

    ensure(i.nextIndex() == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
    ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

   }
  }

  {
   boolean badPrevious = false;
   Object previous = null;
   it.unimi.dsi.fastutil.BidirectionalIterator i;
   java.util.ListIterator j;
   Object I, J;
   KEY_TYPE from = genKey();
   j = new java.util.LinkedList(t.keySet()).listIterator();
   while(j.hasNext()) {
    Object k = j.next();
    if (((Comparable)k).compareTo(KEY2OBJ(from))
 > 0) {
     badPrevious = true;
     j.previous();
     break;
    }
    previous = k;
   }

   i = (it.unimi.dsi.fastutil.BidirectionalIterator)((SORTED_SET)m.keySet()).iterator(from);

   for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting
point " + from + ")");
    ensure(i.hasPrevious() == j.hasPrevious() || badPrevious && (i.hasPrevious() == (previous != null)), "Error (" +
level + ", " + seed + "): divergence in hasPrevious() (iterator with starting point " + from + ")" + badPrevious);
```

```
    if (r.nextFloat() < .8 && i.hasNext()) {
     ensure((I = i.next()).equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next() (" + I + ", " + J + ",
iterator with starting point " + from + ")");
     //System.err.println("Done next " + I + " " + J + "  " + badPrevious);

     badPrevious = false;

     if (r.nextFloat() < 0.5) {
     }
    }
    else
 if (!badPrevious && r.nextFloat() < .2 && i.hasPrevious()) {
     ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

     if (r.nextFloat() < 0.5) {
     }
    }
   }

  }

  /* Now we check that m actually holds that data. */

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after iteration");

  /* Now we select a pair of keys and create a submap. */

  if (! m.isEmpty()) {
   java.util.ListIterator i;
   Object start = m.firstKey(), end = m.firstKey();
   for(i = (java.util.ListIterator)m.keySet().iterator(); i.hasNext() && r.nextFloat() < .3; start = end = i.next());
   for(; i.hasNext() && r.nextFloat() < .95; end = i.next());

   //System.err.println("Checking subMap from " + start + " to " + end + " (level=" + (level+1)
 + ")...");
   testMaps((SORTED_MAP)m.subMap((KEY_CLASS)start, (KEY_CLASS)end), t.subMap(start, end), n, level + 1);

   ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after subMap");
   ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subMap");

   //System.err.println("Checking headMap to " + end + " (level=" + (level+1) + ")...");
   testMaps((SORTED_MAP)m.headMap((KEY_CLASS)end), t.headMap(end), n, level + 1);

   ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after headMap");
   ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after headMap");
```

```
  //System.err.println("Checking tailMap from " + start + " (level=" + (level+1) + ")...");
  testMaps((SORTED_MAP)m.tailMap((KEY_CLASS)start), t.tailMap(start), n, level + 1);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after tailMap");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): !
t.equals(m) after tailMap");
 }


}


private static void test() {
 int n = 1;
 k = new Object[n];
 v = new Object[n];
 nk = new Object[n];
 kt = new KEY_TYPE[n];
 nkt = new KEY_TYPE[n];
 vt = new VALUE_TYPE[n];


 for(int i = 0; i < n; i++) {
#if KEY_CLASS_Object
  k[i] = kt[i] = genKey();
  nk[i] = nkt[i] = genKey();
#else
  k[i] = new KEY_CLASS(kt[i] = genKey());
  nk[i] = new KEY_CLASS(nkt[i] = genKey());
#endif
#if VALUES_REFERENCE
  v[i] = vt[i] = genValue();
#else
  v[i] = new VALUE_CLASS(vt[i] = genValue());
#endif
 }

 SORTED_MAP m = new Singleton(kt[0], vt[0]);
 topMap = m;
 SortedMap t1 = new java.util.TreeMap();
 t1.put(k[0], v[0]);
 SortedMap t = java.util.Collections.unmodifiableSortedMap(t1);

 testMaps(m, t, n, 0);

 System.out.println("Test OK");
 return;
}
```

```
 public static void main(String args[]) {
  if (args.length > 1) r = new java.util.Random(seed = Long.parseLong(args[1]));

  try {
   test();
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed:
 " + seed);
  }
 }

#endif



}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/SortedMaps.drv
No license file was found, but licenses were detected in source scan.

distributed under the <a HREF="http://www.apache.org/licenses/LICENSE-2.0.html">Apache License 2.0</a>.

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/src/overview.html
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package PACKAGE;

import it.unimi.dsi.fastutil.Function;

/** A class providing static methods and objects that do useful things with type-specific functions.
*
* @see it.unimi.dsi.fastutil.Function
* @see java.util.Collections
*/

public final class FUNCTIONS {

 private FUNCTIONS() {}

 /** An immutable class representing
 an empty type-specific function.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific function.
  */

 public static class EmptyFunction KEY_VALUE_GENERIC extends ABSTRACT_FUNCTION
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected EmptyFunction() {}

  @Override
  public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) { return VALUE_NULL; }

  @Override
  public boolean containsKey(final KEY_TYPE k) { return false; }

  @Override
  public VALUE_GENERIC_TYPE defaultReturnValue()  { return VALUE_NULL; }

  @Override
  public void defaultReturnValue(final VALUE_GENERIC_TYPE defRetValue)  { throw new
UnsupportedOperationException(); }

  @Override
  public int size() { return 0; }

  @Override
  public void clear() {}

  @Override
```

```
 public Object clone() { return EMPTY_FUNCTION; }

 @Override
 public int hashCode() { return 0; }

 @Override
 public
boolean equals(final Object o) {
  if (! (o instanceof Function)) return false;
  return ((Function<?,?>)o).size() == 0;
 }

 @Override
 public String toString() { return "{}"; }

 private Object readResolve() { return EMPTY_FUNCTION; }
}


/** An empty type-specific function (immutable). It is serializable and cloneable. */

SUPPRESS_WARNINGS_KEY_VALUE_RAWTYPES
public static final EmptyFunction EMPTY_FUNCTION = new EmptyFunction();


/** An immutable class representing a type-specific singleton function. Note
 *  that the default return value is still settable.
 *
 * <p>Note that albeit the function is immutable, its default return value may be changed.
 *
 * <p>This class may be useful to implement your own in case you subclass
 * a type-specific function.
 */

 public static class Singleton KEY_VALUE_GENERIC extends ABSTRACT_FUNCTION
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable {

 private static final long serialVersionUID = -7046029254386353129L;

 protected
final KEY_GENERIC_TYPE key;
 protected final VALUE_GENERIC_TYPE value;

 protected Singleton(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value) {
  this.key = key;
  this.value = value;
 }
```

```
  @Override
  public boolean containsKey(final KEY_TYPE k) { return KEY_EQUALS(key, k); }

  @Override
  public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) { return KEY_EQUALS(key, k) ? value :
defRetValue; }

  @Override
  public int size() { return 1; }

  @Override
  public Object clone() { return this; }
 }

 /** Returns a type-specific immutable function containing only the specified pair.
  * The returned function is serializable and cloneable.
  *
  * <p>Note that albeit the returned function is immutable, its default return value may be changed.
  *
  * @param key the only key of the returned function.
  * @param value the only value of the returned function.
  * @return a type-specific immutable function containing just the pair {@code &lt;key,value&gt;}.

 */

 public static KEY_VALUE_GENERIC FUNCTION KEY_VALUE_GENERIC singleton(final
KEY_GENERIC_TYPE key, VALUE_GENERIC_TYPE value) {
  return new Singleton KEY_VALUE_GENERIC_DIAMOND(key, value);
 }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

 /** Returns a type-specific immutable function containing only the specified pair. The returned function is
serializable and cloneable.
  *
  * <p>Note that albeit the returned function is immutable, its default return value may be changed.
  *
  * @param key the only key of the returned function.
  * @param value the only value of the returned function.
  * @return a type-specific immutable function containing just the pair {@code &lt;key,value&gt;}.
  */

 public static KEY_VALUE_GENERIC FUNCTION KEY_VALUE_GENERIC singleton(final
KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS value) {
  return new Singleton KEY_VALUE_GENERIC_DIAMOND(KEY_CLASS2TYPE(key),
VALUE_CLASS2TYPE(value));
 }
```

```
#endif


 /** A synchronized wrapper class for functions.
 */

 public static class SynchronizedFunction KEY_VALUE_GENERIC implements FUNCTION
KEY_VALUE_GENERIC, java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final FUNCTION KEY_VALUE_GENERIC function;
  protected final Object sync;

  protected SynchronizedFunction(final FUNCTION KEY_VALUE_GENERIC f, final Object sync) {
   if (f == null) throw new NullPointerException();
   this.function = f;
   this.sync = sync;
  }

  protected SynchronizedFunction(final FUNCTION KEY_VALUE_GENERIC f) {
   if (f == null) throw new NullPointerException();
   this.function = f;
   this.sync = this;
  }

#ifdef JDK_PRIMITIVE_FUNCTION

#if KEY_WIDENED
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
#endif
  @Override
  public VALUE_GENERIC_TYPE_WIDENED
JDK_PRIMITIVE_FUNCTION_APPLY(KEY_GENERIC_TYPE_WIDENED operand) { synchronized(sync) {
return function.JDK_PRIMITIVE_FUNCTION_APPLY(operand);
 } }

#endif

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
#endif
  @Override
  public VALUE_GENERIC_CLASS apply(final KEY_GENERIC_CLASS key) { synchronized (sync) { return
```

```
function.apply(key); } }

  @Override
  public int size() { synchronized(sync) { return function.size(); } }

  @Override
  public VALUE_GENERIC_TYPE defaultReturnValue()  { synchronized(sync) { return
function.defaultReturnValue(); } }

  @Override
  public void defaultReturnValue(final VALUE_GENERIC_TYPE defRetValue)  { synchronized(sync) {
function.defaultReturnValue(defRetValue); } }

  @Override
  public boolean containsKey(final KEY_TYPE k) { synchronized(sync) { return function.containsKey(k); } }

#if KEYS_PRIMITIVE

  @Deprecated
  @Override
  public boolean containsKey(final Object k) { synchronized(sync) { return function.containsKey(k); } }

#endif

  @Override
  public VALUE_GENERIC_TYPE put(final
 KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) { synchronized(sync) { return function.put(k, v); }
}

  @Override
  public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) { synchronized(sync) { return
function.GET_VALUE(k); } }

  @Override
  public VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) { synchronized(sync) { return
function.REMOVE_VALUE(k); } }

  @Override
  public void clear() { synchronized(sync) { function.clear(); } }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS put(final KEY_GENERIC_CLASS k, final VALUE_GENERIC_CLASS v) {
synchronized(sync) { return function.put(k, v); } }
```

```
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS get(final Object k) { synchronized(sync) { return function.get(k); } }

  /** {@inheritDoc}

 * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS remove(final Object k) { synchronized(sync) { return function.remove(k); } }

#endif

  @Override
  public int hashCode() { synchronized(sync) { return function.hashCode(); } }

  @Override
  public boolean equals(final Object o) {
   if (o == this) return true;
   synchronized(sync) { return function.equals(o); }
  }

  @Override
  public String toString() { synchronized(sync) { return function.toString(); } }

  private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
   synchronized(sync) { s.defaultWriteObject(); }
  }
 }

 /** Returns a synchronized type-specific function backed by the given type-specific function.
  *
  * @param f the function to be wrapped in a synchronized function.
  * @return a synchronized view of the specified function.
  * @see java.util.Collections#synchronizedMap(java.util.Map)
  */
 public static
 KEY_VALUE_GENERIC FUNCTION KEY_VALUE_GENERIC synchronize(final FUNCTION
KEY_VALUE_GENERIC f) { return new SynchronizedFunction KEY_VALUE_GENERIC_DIAMOND(f); }

 /** Returns a synchronized type-specific function backed by the given type-specific function, using an assigned
object to synchronize.
  *
  * @param f the function to be wrapped in a synchronized function.
  * @param sync an object that will be used to synchronize the access to the function.
```

```
  * @return a synchronized view of the specified function.
  * @see java.util.Collections#synchronizedMap(java.util.Map)
  */

 public static KEY_VALUE_GENERIC FUNCTION KEY_VALUE_GENERIC synchronize(final FUNCTION
KEY_VALUE_GENERIC f, final Object sync) { return new SynchronizedFunction
KEY_VALUE_GENERIC_DIAMOND(f, sync); }



 /** An unmodifiable wrapper class for functions. */

 public static class UnmodifiableFunction KEY_VALUE_GENERIC extends ABSTRACT_FUNCTION
KEY_VALUE_GENERIC implements java.io.Serializable {

  private static final
 long serialVersionUID = -7046029254386353129L;

  protected final FUNCTION KEY_VALUE_GENERIC function;

  protected UnmodifiableFunction(final FUNCTION KEY_VALUE_GENERIC f) {
   if (f == null) throw new NullPointerException();
   this.function = f;
  }

  @Override
  public int size() { return function.size(); }

  @Override
  public VALUE_GENERIC_TYPE defaultReturnValue()  { return function.defaultReturnValue(); }

  @Override
  public void defaultReturnValue(final VALUE_GENERIC_TYPE defRetValue)  { throw new
UnsupportedOperationException(); }

  @Override
  public boolean containsKey(final KEY_TYPE k) { return function.containsKey(k); }

  @Override
  public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) { throw
new UnsupportedOperationException(); }

  @Override
  public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) { return function.GET_VALUE(k); }

  @Override
  public VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) { throw new
UnsupportedOperationException();
```

```
 }

  @Override
  public void clear() { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS put(final KEY_GENERIC_CLASS k, final VALUE_GENERIC_CLASS v) {
throw new UnsupportedOperationException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS get(final Object k) { return function.get(k); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS remove(final Object k) { throw new UnsupportedOperationException(); }
#endif

  @Override
  public int hashCode() { return function.hashCode(); }

  @Override
  public boolean equals(Object o) { return o == this || function.equals(o); }

  @Override
  public
 String toString() { return function.toString(); }
 }

 /** Returns an unmodifiable type-specific function backed by the given type-specific function.
  *
  * @param f the function to be wrapped in an unmodifiable function.
  * @return an unmodifiable view of the specified function.
  * @see java.util.Collections#unmodifiableMap(java.util.Map)
  */
 public static KEY_VALUE_GENERIC FUNCTION KEY_VALUE_GENERIC unmodifiable(final FUNCTION
KEY_VALUE_GENERIC f) { return new UnmodifiableFunction KEY_VALUE_GENERIC_DIAMOND(f); }


#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
```

```
 /** An adapter for mapping generic total functions to partial primitive functions. */

 public static class PrimitiveFunction KEY_VALUE_GENERIC implements FUNCTION
KEY_VALUE_GENERIC {
  protected final java.util.function.Function<? super KEY_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> function;

  protected PrimitiveFunction(java.util.function.Function<? super KEY_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> function)
 {
   this.function = function;
  }

#if KEYS_PRIMITIVE
  @Override
  public boolean containsKey(KEY_GENERIC_TYPE key) { return function.apply(KEY2OBJ(key)) != null; }
#endif

  SUPPRESS_WARNINGS_KEY_UNCHECKED
#if KEYS_PRIMITIVE
  @Deprecated
#endif
  @Override
  public boolean containsKey(Object key) {
#if KEYS_PRIMITIVE
   if (key == null) return false;
#endif
   return function.apply(KEY_CLASS_CAST (key)) != null;
  }

  SUPPRESS_WARNINGS_KEY_UNCHECKED
  @Override
  public VALUE_GENERIC_TYPE GET_VALUE(KEY_TYPE key) {
   VALUE_GENERIC_CLASS v = function.apply(KEY_GENERIC_CAST KEY2OBJ(key));
#if VALUES_PRIMITIVE
   if (v == null) return defaultReturnValue();
#else
   if (v == null) return null;
#endif
   return VALUE_CLASS2TYPE(v);
  }

  SUPPRESS_WARNINGS_KEY_UNCHECKED
  @Deprecated
  @Override
  public VALUE_GENERIC_CLASS get(Object key) {
#if KEYS_PRIMITIVE
   if (key == null) return null;
```

```
#endif
  return function.apply(KEY_CLASS_CAST key);
 }

 @Deprecated
 @Override
 public
 VALUE_GENERIC_CLASS put(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS value) {
throw new UnsupportedOperationException(); }
 }

 /** Returns a (partial) type-specific function based on the given total generic function.
  * <p>The returned function contains all keys which are not mapped to {@code null}. If the function already
  * is a primitive function, it is returned without changes.
  * <p><strong>Warning</strong>: If the given function is a &ldquo;widened&rdquo; primitive function (e.g. an
  * {@code Int2IntFunction} given to {@code Short2ShortFunctions}), it still is wrapped into a proxy,
  * decreasing performance.
  *
  * @param f the function to be converted to a type-specific function.
  * @return a primitive view of the specified function.
  * @throws NullPointerException if {@code f} is null.
  * @see PrimitiveFunction
  * @since 8.1.0
  */
 SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
 public static KEY_VALUE_GENERIC FUNCTION KEY_VALUE_GENERIC primitive(final
java.util.function.Function<?
 super KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> f) {
  java.util.Objects.requireNonNull(f);

  if (f instanceof FUNCTION) return (FUNCTION KEY_VALUE_GENERIC) f;
#if defined JDK_PRIMITIVE_FUNCTION
  if (f instanceof JDK_PRIMITIVE_FUNCTION)
#if KEYS_PRIMITIVE
#if VALUE_WIDENED
  return key -> VALUE_NARROWING(((JDK_PRIMITIVE_FUNCTION KEY_VALUE_GENERIC)
f).JDK_PRIMITIVE_FUNCTION_APPLY(key));
#else
  return ((JDK_PRIMITIVE_FUNCTION KEY_VALUE_GENERIC) f)::JDK_PRIMITIVE_FUNCTION_APPLY;
#endif
#else
  return key -> VALUE_NARROWING(((JDK_PRIMITIVE_FUNCTION KEY_VALUE_GENERIC)
f).JDK_PRIMITIVE_FUNCTION_APPLY((K)(key)));
#endif
#endif
  return new PrimitiveFunction KEY_VALUE_GENERIC_DIAMOND(f);
 }
#endif
```

}

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/Functions.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import it.unimi.dsi.fastutil.Hash;
import it.unimi.dsi.fastutil.HashCommon;
import static it.unimi.dsi.fastutil.HashCommon.arraySize;
import static it.unimi.dsi.fastutil.HashCommon.maxFill;

import java.util.Arrays;
import java.util.Collection;
import java.util.Iterator;
import java.util.NoSuchElementException;

#ifdef
 Linked

#if KEYS_REFERENCE
import java.util.Comparator;
#endif

/**  A type-specific linked hash set with with a fast, small-footprint implementation.
*
* <p>Instances of this class use a hash table to represent a set. The table is
* filled up to a specified <em>load factor</em>, and then doubled in size to
* accommodate new entries. If the table is emptied below <em>one fourth</em>
```

```
* of the load factor, it is halved in size; however, the table is never reduced to a
* size smaller than that at creation time: this approach makes it
* possible to create sets with a large capacity in which insertions and
* deletions do not cause immediately rehashing. Moreover, halving is
* not performed when deleting entries from an iterator, as it would interfere
* with the iteration process.
*
* <p>Note that {@link #clear()} does not modify the hash table size.
* Rather, a family of {@linkplain #trim() trimming
* methods} lets you control the size of the table; this is
 particularly useful
* if you reuse instances of this class.
*
* <p>Iterators generated by this set will enumerate elements in the same order in which they
* have been added to the set (addition of elements already present
* in the set does not change the iteration order). Note that this order has nothing in common with the natural
* order of the keys. The order is kept by means of a doubly linked list, represented
* <i>via</i> an array of longs parallel to the table.
*
* <p>This class implements the interface of a sorted set, so to allow easy
* access of the iteration order: for instance, you can get the first element
* in iteration order with {@code first()} without having to create an
* iterator; however, this class partially violates the {@link java.util.SortedSet}
* contract because all subset methods throw an exception and {@link
* #comparator()} returns always {@code null}.
*
* <p>Additional methods, such as {@code addAndMoveToFirst()}, make it easy
* to use instances
 of this class as a cache (e.g., with LRU policy).
*
* <p>The iterators provided by this class are type-specific {@linkplain
* java.util.ListIterator list iterators}, and can be started at any
* element <em>which is in the set</em> (if the provided element
* is not in the set, a {@link NoSuchElementException} exception will be thrown).
* If, however, the provided element is not the first or last element in the
* set, the first access to the list index will require linear time, as in the worst case
* the entire set must be scanned in iteration order to retrieve the positional
* index of the starting element. If you use just the methods of a type-specific {@link
it.unimi.dsi.fastutil.BidirectionalIterator},
* however, all operations will be performed in constant time.
*
* @see Hash
* @see HashCommon
*/

public class OPEN_HASH_SET KEY_GENERIC extends ABSTRACT_SORTED_SET KEY_GENERIC
implements java.io.Serializable, Cloneable, Hash {
```

```
#else

#ifdef Custom

/** A type-specific
 hash set with a fast, small-footprint implementation whose {@linkplain it.unimi.dsi.fastutil.Hash.Strategy hashing
strategy}
* is specified at creation time.
*
* <p>Instances of this class use a hash table to represent a set. The table is
* filled up to a specified <em>load factor</em>, and then doubled in size to
* accommodate new entries. If the table is emptied below <em>one fourth</em>
* of the load factor, it is halved in size; however, the table is never reduced to a
* size smaller than that at creation time: this approach makes it
* possible to create sets with a large capacity in which insertions and
* deletions do not cause immediately rehashing. Moreover, halving is
* not performed when deleting entries from an iterator, as it would interfere
* with the iteration process.
*
* <p>Note that {@link #clear()} does not modify the hash table size.
* Rather, a family of {@linkplain #trim() trimming
* methods} lets you control the size of the table; this is particularly
 useful
* if you reuse instances of this class.
*
* @see Hash
* @see HashCommon
*/

public class OPEN_HASH_SET KEY_GENERIC extends ABSTRACT_SET KEY_GENERIC implements
java.io.Serializable, Cloneable, Hash {

#else

/**  A type-specific hash set with with a fast, small-footprint implementation.
*
* <p>Instances of this class use a hash table to represent a set. The table is
* filled up to a specified <em>load factor</em>, and then doubled in size to
* accommodate new entries. If the table is emptied below <em>one fourth</em>
* of the load factor, it is halved in size; however, the table is never reduced to a
* size smaller than that at creation time: this approach makes it
* possible to create sets with a large capacity in which insertions and
* deletions do not cause immediately rehashing. Moreover, halving is
* not performed when deleting entries from an iterator, as it would interfere
* with the iteration process.
*
* <p>Note that {@link #clear()} does not modify
```

```
 the hash table size.
* Rather, a family of {@linkplain #trim() trimming
* methods} lets you control the size of the table; this is particularly useful
* if you reuse instances of this class.
*
* @see Hash
* @see HashCommon
*/

public class OPEN_HASH_SET KEY_GENERIC extends ABSTRACT_SET KEY_GENERIC implements
java.io.Serializable, Cloneable, Hash {

#endif

#endif

 private static final long serialVersionUID = 0L;
 private static final boolean ASSERTS = ASSERTS_VALUE;

 /** The array of keys. */
 protected transient KEY_GENERIC_TYPE[] key;

 /** The mask for wrapping a position counter. */
 protected transient int mask;

 /** Whether this set contains the null key. */
 protected transient boolean containsNull;

#ifdef Custom
 /** The hash strategy of this custom set. */
 protected STRATEGY KEY_SUPER_GENERIC strategy;
#endif

#ifdef Linked
 /** The index of the first entry in iteration order. It is valid iff {@link #size} is nonzero; otherwise, it contains -1. */
 protected transient
 int first = -1;
 /** The index of the last entry in iteration order. It is valid iff {@link #size} is nonzero; otherwise, it contains -1. */
 protected transient int last = -1;
 /** For each entry, the next and the previous entry in iteration order,
  * stored as {@code ((prev & 0xFFFFFFFFL) << 32) | (next & 0xFFFFFFFFL)}.
  * The first entry contains predecessor -1, and the last entry
  * contains successor -1. */
 protected transient long[] link;
#endif

 /** The current table size. Note that an additional element is allocated for storing the null key. */
 protected transient int n;
```

```
 /** Threshold after which we rehash. It must be the table size times {@link #f}. */
 protected transient int maxFill;

 /** We never resize below this threshold, which is the construction-time {#n}. */
 protected final transient int minN;

 /** Number of entries in the set (including the null key, if present). */
 protected int size;

 /** The acceptable load factor. */
 protected final float f;

#ifdef
 Custom
 /** Creates a new hash set.
  *
  * <p>The actual table size will be the least power of two greater than {@code expected}/{@code f}.
  *
  * @param expected the expected number of elements in the hash set.
  * @param f the load factor.
  * @param strategy the strategy.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public OPEN_HASH_SET(final int expected, final float f, final STRATEGY KEY_SUPER_GENERIC strategy) {
  this.strategy = strategy;
#else
 /** Creates a new hash set.
  *
  * <p>The actual table size will be the least power of two greater than {@code expected}/{@code f}.
  *
  * @param expected the expected number of elements in the hash set.
  * @param f the load factor.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public OPEN_HASH_SET(final int expected, final float f) {
#endif
  if (f <= 0 || f >= 1) throw new IllegalArgumentException("Load factor must be greater than 0 and smaller than 1");
  if (expected < 0) throw new IllegalArgumentException("The expected number of elements
 must be nonnegative");

  this.f = f;

  minN = n = arraySize(expected, f);
  mask = n - 1;
  maxFill = maxFill(n, f);
  key = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[n + 1];
#ifdef Linked
```

```
  link = new long[n + 1];
#endif
 }


#ifdef Custom
 /** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor.
  *
  * @param expected the expected number of elements in the hash set.
  * @param strategy the strategy.
  */

 public OPEN_HASH_SET(final int expected, final STRATEGY KEY_SUPER_GENERIC strategy) {
  this(expected, DEFAULT_LOAD_FACTOR, strategy);
 }

#else
 /** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor.
  *
  * @param expected the expected number of elements in the hash set.
  */

 public OPEN_HASH_SET(final int expected) {
  this(expected, DEFAULT_LOAD_FACTOR);
 }

#endif


#ifdef Custom
 /** Creates a new hash set with initial expected {@link Hash#DEFAULT_INITIAL_SIZE} elements
  * and {@link Hash#DEFAULT_LOAD_FACTOR} as load factor.

 * @param strategy the strategy.
  */

 public OPEN_HASH_SET(final STRATEGY KEY_SUPER_GENERIC strategy) {
 this(DEFAULT_INITIAL_SIZE, DEFAULT_LOAD_FACTOR, strategy);
 }
#else
 /** Creates a new hash set with initial expected {@link Hash#DEFAULT_INITIAL_SIZE} elements
  * and {@link Hash#DEFAULT_LOAD_FACTOR} as load factor.
  */

 public OPEN_HASH_SET() {
  this(DEFAULT_INITIAL_SIZE, DEFAULT_LOAD_FACTOR);
 }
#endif
```

```
#ifdef Custom
 /** Creates a new hash set copying a given collection.
  *
  * @param c a {@link Collection} to be copied into the new hash set.
  * @param f the load factor.
  * @param strategy the strategy.
  */

 public OPEN_HASH_SET(final Collection<? extends KEY_GENERIC_CLASS> c, final float f, final STRATEGY
KEY_SUPER_GENERIC strategy) {
 this(c.size(), f, strategy);
  addAll(c);
 }
#else
 /** Creates a new hash set copying a given collection.
  *
  * @param c a {@link Collection} to be copied into the new hash set.
  * @param f the load factor.
  */

 public OPEN_HASH_SET(final
 Collection<? extends KEY_GENERIC_CLASS> c, final float f) {
  this(c.size(), f);
  addAll(c);
 }
#endif



#ifdef Custom
 /** Creates a new hash set  with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
  * copying a given collection.
  *
  * @param c a {@link Collection} to be copied into the new hash set.
  * @param strategy the strategy.
  */

 public OPEN_HASH_SET(final Collection<? extends KEY_GENERIC_CLASS> c, final STRATEGY
KEY_SUPER_GENERIC strategy) {
  this(c, DEFAULT_LOAD_FACTOR, strategy);
 }

#else
 /** Creates a new hash set  with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
  * copying a given collection.
  *
```

```
  * @param c a {@link Collection} to be copied into the new hash set.
  */

 public OPEN_HASH_SET(final Collection<? extends KEY_GENERIC_CLASS> c) {
  this(c, DEFAULT_LOAD_FACTOR);
 }
#endif


#ifdef Custom
 /** Creates a new hash set copying a given type-specific collection.
  *
  * @param c a type-specific collection to be copied into the new
 hash set.
  * @param f the load factor.
  * @param strategy the strategy.
  */

 public OPEN_HASH_SET(final COLLECTION KEY_EXTENDS_GENERIC c, final float f, STRATEGY
KEY_SUPER_GENERIC strategy) {
 this(c.size(), f, strategy);
  addAll(c);
 }
#else
 /** Creates a new hash set copying a given type-specific collection.
  *
  * @param c a type-specific collection to be copied into the new hash set.
  * @param f the load factor.
  */

 public OPEN_HASH_SET(final COLLECTION KEY_EXTENDS_GENERIC c, final float f) {
  this(c.size(), f);
  addAll(c);
 }
#endif


#ifdef Custom
 /** Creates a new hash set  with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
  * copying a given type-specific collection.
  *
  * @param c a type-specific collection to be copied into the new hash set.
  * @param strategy the strategy.
  */

 public OPEN_HASH_SET(final COLLECTION KEY_EXTENDS_GENERIC c, final STRATEGY
KEY_SUPER_GENERIC strategy) {
  this(c, DEFAULT_LOAD_FACTOR, strategy);
```

```
 }
#else
 /** Creates a new hash
 set  with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
  * copying a given type-specific collection.
  *
  * @param c a type-specific collection to be copied into the new hash set.
  */

 public OPEN_HASH_SET(final COLLECTION KEY_EXTENDS_GENERIC c) {
  this(c, DEFAULT_LOAD_FACTOR);
 }
#endif


#ifdef Custom
 /** Creates a new hash set using elements provided by a type-specific iterator.
  *
  * @param i a type-specific iterator whose elements will fill the set.
  * @param f the load factor.
  * @param strategy the strategy.
  */

 public OPEN_HASH_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i, final float f, final
STRATEGY KEY_SUPER_GENERIC strategy) {
  this(DEFAULT_INITIAL_SIZE, f, strategy);
  while(i.hasNext()) add(i.NEXT_KEY());
 }
#else
 /** Creates a new hash set using elements provided by a type-specific iterator.
  *
  * @param i a type-specific iterator whose elements will fill the set.
  * @param f the load factor.
  */

 public OPEN_HASH_SET(final STD_KEY_ITERATOR
 KEY_EXTENDS_GENERIC i, final float f) {
  this(DEFAULT_INITIAL_SIZE, f);
  while(i.hasNext()) add(i.NEXT_KEY());
 }
#endif

#ifdef Custom
 /** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor using elements
provided by a type-specific iterator.
  *
  * @param i a type-specific iterator whose elements will fill the set.
  * @param strategy the strategy.
```

```
  */

 public OPEN_HASH_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i, final STRATEGY
KEY_SUPER_GENERIC strategy) {
  this(i, DEFAULT_LOAD_FACTOR, strategy);
 }
#else
 /** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor using elements
provided by a type-specific iterator.
  *
  * @param i a type-specific iterator whose elements will fill the set.
  */

 public OPEN_HASH_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i) {
  this(i, DEFAULT_LOAD_FACTOR);
 }
#endif



#if KEYS_PRIMITIVE

#ifdef Custom
 /** Creates a new hash set using elements provided by an iterator.
  *
  * @param
 i an iterator whose elements will fill the set.
  * @param f the load factor.
  * @param strategy the strategy.
  */

 public OPEN_HASH_SET(final Iterator<?> i, final float f, final STRATEGY KEY_SUPER_GENERIC strategy) {
  this(ITERATORS.AS_KEY_ITERATOR(i), f, strategy);
 }
#else
 /** Creates a new hash set using elements provided by an iterator.
  *
  * @param i an iterator whose elements will fill the set.
  * @param f the load factor.
  */

 public OPEN_HASH_SET(final Iterator<?> i, final float f) {
  this(ITERATORS.AS_KEY_ITERATOR(i), f);
 }

#endif
```

```
#ifdef Custom
 /** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor using elements
provided by an iterator.
  *
  * @param i an iterator whose elements will fill the set.
  * @param strategy the strategy.
  */

 public OPEN_HASH_SET(final Iterator<?> i, final STRATEGY KEY_SUPER_GENERIC strategy) {
  this(ITERATORS.AS_KEY_ITERATOR(i), strategy);
 }
#else
 /** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR}
 as load factor using elements provided by an iterator.
  *
  * @param i an iterator whose elements will fill the set.
  */

 public OPEN_HASH_SET(final Iterator<?> i) {
  this(ITERATORS.AS_KEY_ITERATOR(i));
 }
#endif

#endif



#ifdef Custom
 /** Creates a new hash set and fills it with the elements of a given array.
  *
  * @param a an array whose elements will be used to fill the set.
  * @param offset the first element to use.
  * @param length the number of elements to use.
  * @param f the load factor.
  * @param strategy the strategy.
  */

 public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length, final float f, final
STRATEGY KEY_SUPER_GENERIC strategy) {
 this(length < 0 ? 0 : length, f, strategy);
  ARRAYS.ensureOffsetLength(a, offset, length);
  for(int i = 0; i < length; i++) add(a[offset + i]);
 }
#else
 /** Creates a new hash set and fills it with the elements of a given array.
  *
  * @param a an array whose elements will be used to
```

```
 fill the set.
  * @param offset the first element to use.
  * @param length the number of elements to use.
  * @param f the load factor.
  */

 public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length, final float f) {
  this(length < 0 ? 0 : length, f);
  ARRAYS.ensureOffsetLength(a, offset, length);
  for(int i = 0; i < length; i++) add(a[offset + i]);
 }
#endif

#ifdef Custom
 /** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor and fills it with the
elements of a given array.
  *
  * @param a an array whose elements will be used to fill the set.
  * @param offset the first element to use.
  * @param length the number of elements to use.
  * @param strategy the strategy.
  */

 public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length, final STRATEGY
KEY_SUPER_GENERIC strategy) {
  this(a, offset, length, DEFAULT_LOAD_FACTOR, strategy);
 }
#else
 /** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR}
 as load factor and fills it with the elements of a given array.
  *
  * @param a an array whose elements will be used to fill the set.
  * @param offset the first element to use.
  * @param length the number of elements to use.
  */

 public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length) {
  this(a, offset, length, DEFAULT_LOAD_FACTOR);
 }
#endif

#ifdef Custom
 /** Creates a new hash set copying the elements of an array.
  *
  * @param a an array to be copied into the new hash set.
  * @param f the load factor.
  * @param strategy the strategy.
  */
```

```
 public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final float f, final STRATEGY
KEY_SUPER_GENERIC strategy) {
  this(a, 0, a.length, f, strategy);
 }
#else
 /** Creates a new hash set copying the elements of an array.
  *
  * @param a an array to be copied into the new hash set.
  * @param f the load factor.
  */

 public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final float f) {
  this(a, 0, a.length,
 f);
 }
#endif

#ifdef Custom
 /** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
  * copying the elements of an array.
  *
  * @param a an array to be copied into the new hash set.
  * @param strategy the strategy.
  */

 public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final STRATEGY KEY_SUPER_GENERIC
strategy) {
  this(a, DEFAULT_LOAD_FACTOR, strategy);
 }
#else
 /** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
  * copying the elements of an array.
  *
  * @param a an array to be copied into the new hash set.
  */

 public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a) {
  this(a, DEFAULT_LOAD_FACTOR);
 }
#endif


#ifdef Custom
 /** Returns the hashing strategy.
  *
  * @return the hashing strategy of this custom hash set.
  */
```

```
 public STRATEGY KEY_SUPER_GENERIC strategy() {
  return strategy;
 }
#endif

 private int realSize() {
  return containsNull ? size - 1 : size;
 }

 private void ensureCapacity(final int capacity) {
  final
 int needed = arraySize(capacity, f);
  if (needed > n) rehash(needed);
 }

 private void tryCapacity(final long capacity) {
  final int needed = (int)Math.min(1 << 30, Math.max(2, HashCommon.nextPowerOfTwo((long)Math.ceil(capacity /
f))));
  if (needed > n) rehash(needed);
 }

#if KEYS_PRIMITIVE
 @Override
 public boolean addAll(COLLECTION c) {
  if (f <= .5) ensureCapacity(c.size()); // The resulting collection will be sized for c.size() elements
  else tryCapacity(size() + c.size()); // The resulting collection will be tentatively sized for size() + c.size() elements
  return super.addAll(c);
 }
#endif

 @Override
 public boolean addAll(Collection<? extends KEY_GENERIC_CLASS> c) {
  // The resulting collection will be at least c.size() big
  if (f <= .5) ensureCapacity(c.size()); // The resulting collection will be sized for c.size() elements
  else tryCapacity(size() + c.size()); // The resulting collection will be tentatively sized for size() + c.size() elements
  return super.addAll(c);
 }


 @Override
 public
 boolean add(final KEY_GENERIC_TYPE k) {
  int pos;

  if (KEY_EQUALS_NULL(k)) {
   if (containsNull) return false;
#ifdef Linked
```

```
    pos = n;
#endif
    containsNull = true;
#ifdef Custom
    key[n] = k;
#endif
   }
   else {
    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = this.key;

    // The starting point.
    if (! KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) {
     if (KEY_EQUALS_NOT_NULL(curr, k)) return false;
     while(! KEY_IS_NULL(curr = key[pos = (pos + 1) & mask]))
      if (KEY_EQUALS_NOT_NULL(curr, k)) return false;
    }
    key[pos] = k;
   }

#ifdef Linked
   if (size == 0) {
    first = last = pos;
    // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
    link[pos] = -1L;
   }
   else {
    SET_NEXT(link[last], pos);
    SET_UPPER_LOWER(link[pos], last, -1);
    last = pos;
   }
#endif

   if (size++ >= maxFill) rehash(arraySize(size + 1, f));
   if (ASSERTS) checkTable();
   return true;
  }

#if KEY_CLASS_Object
  /** Add a random
  element if not present, get the existing value if already present.
   *
   * This is equivalent to (but faster than) doing a:
   * <pre>
   * K exist = set.get(k);
   * if (exist == null) {
   *   set.add(k);
   *   exist = k;
```

```
  * }
  * </pre>
  */
 public KEY_GENERIC_TYPE addOrGet(final KEY_GENERIC_TYPE k) {
  int pos;

  if (KEY_EQUALS_NULL(k)) {
   if (containsNull) return key [n];
#ifdef Linked
   pos = n;
#endif
   containsNull = true;
#ifdef Custom
   key [n] = k;
#endif
  }
  else {
   KEY_GENERIC_TYPE curr;
   final KEY_GENERIC_TYPE[] key = this.key;

   // The starting point.
   if (! KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) {
    if (KEY_EQUALS_NOT_NULL(curr, k)) return curr;
    while(! KEY_IS_NULL(curr = key[pos = (pos + 1) & mask]))
     if (KEY_EQUALS_NOT_NULL(curr, k)) return curr;
   }
   key[pos] = k;
  }

#ifdef Linked
  if (size == 0) {
   first = last = pos;
   // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
   link[pos] = -1L;
  }
  else {
   SET_NEXT(link[last],
 pos);
   SET_UPPER_LOWER(link[pos], last, -1);
   last = pos;
  }
#endif

  if (size++ >= maxFill) rehash(arraySize(size + 1, f));
  if (ASSERTS) checkTable();
  return k;
 }
#endif
```

```
/** Shifts left entries with the specified hash code, starting at the specified position,
 * and empties the resulting free entry.
 *
 * @param pos a starting position.
 */
protected final void shiftKeys(int pos) {
 // Shift entries with the same hash.
 int last, slot;
 KEY_GENERIC_TYPE curr;
 final KEY_GENERIC_TYPE[] key = this.key;

 for(;;) {
  pos = ((last = pos) + 1) & mask;

  for(;;) {
   if (KEY_IS_NULL(curr = key[pos])) {
    key[last] = KEY_NULL;
    return;
   }
   slot = KEY2INTHASH(curr) & mask;
   if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
   pos = (pos + 1) & mask;
  }

  key[last] = curr;
#ifdef Linked
  fixPointers(pos, last);
#endif
 }
}

private boolean removeEntry(final int pos) {
 size--;
#ifdef Linked
 fixPointers(pos);
#endif
 shiftKeys(pos);
 if
(n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
 return true;
}

private boolean removeNullEntry() {
 containsNull = false;
 key[n] = KEY_NULL;
 size--;
#ifdef Linked
```

```
 fixPointers(n);
#endif
 if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
 return true;
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public boolean remove(final KEY_TYPE k) {
 if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) {
  if (containsNull) return removeNullEntry();
  return false;
 }

 KEY_GENERIC_TYPE curr;
 final KEY_GENERIC_TYPE[] key = this.key;
 int pos;

 // The starting point.
 if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return false;
 if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return removeEntry(pos);

 while(true) {
  if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return false;
  if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return removeEntry(pos);
 }
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public
boolean contains(final KEY_TYPE k) {
 if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) return containsNull;

 KEY_GENERIC_TYPE curr;
 final KEY_GENERIC_TYPE[] key = this.key;
 int pos;

 // The starting point.
 if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return false;
 if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return true;

 while(true) {
  if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return false;
  if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return true;
 }
}
```

```
#if KEY_CLASS_Object
 /** Returns the element of this set that is equal to the given key, or {@code null}.
  * @return the element of this set that is equal to the given key, or {@code null}.
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 public K get(final Object k) {
  if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) return key[n]; // This is correct independently of the value
of containsNull and of the set being custom

  KEY_GENERIC_TYPE curr;
  final KEY_GENERIC_TYPE[] key = this.key;
  int
 pos;

  // The starting point.
  if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return null;
  if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return curr;
  // There's always an unused entry.
  while(true) {
   if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return null;
   if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return curr;
  }
 }
#endif

#ifdef Linked

 /** Removes the first key in iteration order.
  * @return the first key.
  * @throws NoSuchElementException is this set is empty.
  */
 public KEY_GENERIC_TYPE REMOVE_FIRST_KEY() {
  if (size == 0) throw new NoSuchElementException();
  final int pos = first;
  // Abbreviated version of fixPointers(pos)
  first = GET_NEXT(link[pos]);
  if (0 <= first) {
   // Special case of SET_PREV(link[first], -1)
   link[first] |= (-1 & 0xFFFFFFFFL) << 32;
  }
  final KEY_GENERIC_TYPE k = key[pos];
  size--;
  if (KEY_EQUALS_NULL(k)) {
   containsNull = false;
   key[n] = KEY_NULL;
  }
  else shiftKeys(pos);
  if (n > minN
```

```
&& size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
 return k;
}

/** Removes the the last key in iteration order.
 * @return the last key.
 * @throws NoSuchElementException is this set is empty.
 */
public KEY_GENERIC_TYPE REMOVE_LAST_KEY() {
 if (size == 0) throw new NoSuchElementException();
 final int pos = last;
 // Abbreviated version of fixPointers(pos)
 last = GET_PREV(link[pos]);
 if (0 <= last) {
  // Special case of SET_NEXT(link[last], -1)
  link[last] |= -1 & 0xFFFFFFFFL;
 }
 final KEY_GENERIC_TYPE k = key[pos];
 size--;
 if (KEY_EQUALS_NULL(k)) {
  containsNull = false;
  key[n] = KEY_NULL;
 }
 else shiftKeys(pos);
 if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
 return k;
}

private void moveIndexToFirst(final int i) {
 if (size == 1 || first == i) return;
 if (last == i) {
  last = GET_PREV(link[i]);
  // Special case of SET_NEXT(link[last], -1);
  link[last] |= -1 & 0xFFFFFFFFL;
 }
 else {
  final
long linki = link[i];
  final int prev = GET_PREV(linki);
  final int next = GET_NEXT(linki);
  COPY_NEXT(link[prev], linki);
  COPY_PREV(link[next], linki);
 }
 SET_PREV(link[first], i);
 SET_UPPER_LOWER(link[i], -1, first);
 first = i;
}
```

```
private void moveIndexToLast(final int i) {
 if (size == 1 ||  last == i) return;
 if (first == i) {
  first = GET_NEXT(link[i]);
  // Special case of SET_PREV(link[first], -1);
  link[first] |= (-1 & 0xFFFFFFFFL) << 32;
 }
 else {
  final long linki = link[i];
  final int prev = GET_PREV(linki);
  final int next = GET_NEXT(linki);
  COPY_NEXT(link[prev], linki);
  COPY_PREV(link[next], linki);
 }
 SET_NEXT(link[last], i);
 SET_UPPER_LOWER(link[i], last, -1);
 last = i;
}

/** Adds a key to the set; if the key is already present, it is moved to the first position of the iteration order.
 *
 * @param k the key.
 * @return true if the key was not present.
 */
public boolean addAndMoveToFirst(final KEY_GENERIC_TYPE
k) {
 int pos;

 if (KEY_EQUALS_NULL(k)) {
  if (containsNull) {
   moveIndexToFirst(n);
   return false;
  }
  containsNull = true;
  pos = n;
 }
 else {
  // The starting point.
  final KEY_GENERIC_TYPE key[] = this.key;
  pos = KEY2INTHASH(k) & mask;

  // There's always an unused entry. TODO
  while(! KEY_IS_NULL(key[pos])) {
   if (KEY_EQUALS_NOT_NULL(k, key[pos])) {
    moveIndexToFirst(pos);
    return false;
   }
```

```
  pos = (pos + 1) & mask;
 }
}

key[pos] = k;

if (size == 0) {
 first = last = pos;
 // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
 link[pos] = -1L;
}
else {
 SET_PREV(link[first], pos);
 SET_UPPER_LOWER(link[pos], -1, first);
 first = pos;
}

if (size++ >= maxFill) rehash(arraySize(size, f));
if (ASSERTS) checkTable();
return true;
}

/** Adds a key to the set; if the key is already present, it is moved to the last position of the iteration order.
 *
 * @param k the key.
 * @return true if the key
was not present.
 */
public boolean addAndMoveToLast(final KEY_GENERIC_TYPE k) {
 int pos;

 if (KEY_EQUALS_NULL(k)) {
  if (containsNull) {
   moveIndexToLast(n);
   return false;
  }
  containsNull = true;
  pos = n;
 }
 else {
  // The starting point.
  final KEY_GENERIC_TYPE key[] = this.key;
  pos = KEY2INTHASH(k) & mask;

  // There's always an unused entry.
  while(! KEY_IS_NULL(key[pos])) {
   if (KEY_EQUALS_NOT_NULL(k, key[pos])) {
    moveIndexToLast(pos);
```

```
    return false;
    }

    pos = (pos + 1) & mask;
   }
  }

  key[pos] = k;

  if (size == 0) {
   first = last = pos;
   // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
   link[pos] = -1L;
  }
  else {
   SET_NEXT(link[last], pos);
   SET_UPPER_LOWER(link[pos], last, -1);
   last = pos;
  }

  if (size++ >= maxFill) rehash(arraySize(size, f));
  if (ASSERTS) checkTable();
  return true;
 }

#endif

 /* Removes all elements from this set.
  *
  * <p>To increase object reuse, this method does
 not change the table size.
  * If you want to reduce the table size, you must use {@link #trim()}.
  *
  */
 @Override
 public void clear() {
  if (size == 0) return;
  size = 0;
  containsNull = false;
  Arrays.fill(key, KEY_NULL);
#ifdef Linked
  first = last = -1;
#endif
 }


 @Override
 public int size() {
```

```
 return size;
}

@Override
public boolean isEmpty() {
 return size == 0;
}

#ifdef Linked

/** Modifies the {@link #link} vector so that the given entry is removed.
 * This method will complete in constant time.
 *
 * @param i the index of an entry.
 */
protected void fixPointers(final int i) {
 if (size == 0) {
  first = last = -1;
  return;
 }
 if (first == i) {
  first = GET_NEXT(link[i]);
  if (0 <= first) {
   // Special case of SET_PREV(link[first], -1)
   link[first] |= (-1 & 0xFFFFFFFFL) << 32;
  }
  return;
 }
 if (last == i) {
  last = GET_PREV(link[i]);
  if (0 <= last) {
   // Special case of SET_NEXT(link[last], -1)
   link[last] |= -1 &
0xFFFFFFFFL;
  }
  return;
 }
 final long linki = link[i];
 final int prev = GET_PREV(linki);
 final int next = GET_NEXT(linki);
 COPY_NEXT(link[prev], linki);
 COPY_PREV(link[next], linki);
}


/** Modifies the {@link #link} vector for a shift from s to d.
 * This method will complete in constant time.
 *
```

```
 * @param s the source position.
 * @param d the destination position.
 */
protected void fixPointers(int s, int d) {
 if (size == 1) {
  first = last = d;
  // Special case of SET(link[d], -1, -1)
  link[d] = -1L;
  return;
 }
 if (first == s) {
  first = d;
  SET_PREV(link[GET_NEXT(link[s])], d);
  link[d] = link[s];
  return;
 }
 if (last == s) {
  last = d;
  SET_NEXT(link[GET_PREV(link[s])], d);
  link[d] = link[s];
  return;
 }
 final long links = link[s];
 final int prev = GET_PREV(links);
 final int next = GET_NEXT(links);
 SET_NEXT(link[prev], d);
 SET_PREV(link[next], d);
 link[d] = links;
}


/** Returns the first element of this set
in iteration order.
 *
 * @return the first element in iteration order.
 */
@Override
public KEY_GENERIC_TYPE FIRST() {
 if (size == 0) throw new NoSuchElementException();
 return key[first];
}

/** Returns the last element of this set in iteration order.
 *
 * @return the last element in iteration order.
 */
@Override
public KEY_GENERIC_TYPE LAST() {
```

```
 if (size == 0) throw new NoSuchElementException();
 return key[last];
 }

 /** {@inheritDoc}
  * <p>This implementation just throws an {@link UnsupportedOperationException}.*/
 @Override
 public SORTED_SET KEY_GENERIC tailSet(KEY_GENERIC_TYPE from) { throw new
UnsupportedOperationException(); }

 /** {@inheritDoc}
  * <p>This implementation just throws an {@link UnsupportedOperationException}.*/
 @Override
 public SORTED_SET KEY_GENERIC headSet(KEY_GENERIC_TYPE to) { throw new
UnsupportedOperationException(); }

 /** {@inheritDoc}
  * <p>This implementation just throws an {@link UnsupportedOperationException}.*/
 @Override
 public
 SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE to) { throw new
UnsupportedOperationException(); }

 /** {@inheritDoc}
  * <p>This implementation just returns {@code null}.*/
 @Override
 public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return null; }


 /** A list iterator over a linked set.
  *
  * <p>This class provides a list iterator over a linked hash set. The constructor runs in constant time.
  */
 private class SetIterator implements KEY_LIST_ITERATOR KEY_GENERIC {
  /** The entry that will be returned by the next call to {@link java.util.ListIterator#previous()} (or {@code null} if
no previous entry exists). */
  int prev = -1;
  /** The entry that will be returned by the next call to {@link java.util.ListIterator#next()} (or {@code null} if no
next entry exists). */
  int next = -1;
  /** The last entry that was returned (or -1 if we did not iterate or used {@link #remove()}). */
  int curr = -1;
  /** The current index (in the sense of
 a {@link java.util.ListIterator}). When -1, we do not know the current index.*/
  int index = -1;

  SetIterator() {
   next = first;
```

```
 index = 0;
}

SetIterator(KEY_GENERIC_TYPE from) {
 if (KEY_EQUALS_NULL(from)) {
  if (OPEN_HASH_SET.this.containsNull) {
   next = GET_NEXT(link[n]);
   prev = n;
   return;
  }
  else throw new NoSuchElementException("The key " + from + " does not belong to this set.");
 }

 if (KEY_EQUALS(key[last], from)) {
  prev = last;
  index = size;
  return;
 }

 // The starting point.
 final KEY_GENERIC_TYPE key[] = OPEN_HASH_SET.this.key;
 int pos = KEY2INTHASH(from) & mask;

 // There's always an unused entry.
 while(! KEY_IS_NULL(key[pos])) {
  if (KEY_EQUALS_NOT_NULL(key[pos], from)) {
   // Note: no valid index known.
   next = GET_NEXT(link[pos]);
   prev = pos;
   return;
  }
  pos = (pos + 1) & mask;
 }
 throw new NoSuchElementException("The key " + from + " does not belong to this set.");
}

@Override
public
boolean hasNext() { return next != -1; }
@Override
public boolean hasPrevious() { return prev != -1; }

@Override
public KEY_GENERIC_TYPE NEXT_KEY() {
 if (! hasNext()) throw new NoSuchElementException();

 curr = next;
 next = GET_NEXT(link[curr]);
```

```
 prev = curr;

 if (index >= 0) index++;
 if (ASSERTS) assert curr == n || ! KEY_IS_NULL(key[curr]) : "Position " + curr + " is not used";
 return key[curr];
}

@Override
public KEY_GENERIC_TYPE PREV_KEY() {
 if (! hasPrevious()) throw new NoSuchElementException();

 curr = prev;
 prev = GET_PREV(link[curr]);
 next = curr;

 if (index >= 0) index--;

 return key[curr];
}

private final void ensureIndexKnown() {
 if (index >= 0) return;
 if (prev == -1) {
  index = 0;
  return;
 }
 if (next == -1) {
  index = size;
  return;
 }
 int pos = first;
 index = 1;
 while(pos != prev) {
  pos = GET_NEXT(link[pos]);
  index++;
 }
}

@Override
public
int nextIndex() {
 ensureIndexKnown();
 return index;
}

@Override
public int previousIndex() {
 ensureIndexKnown();
```

```
 return index - 1;
}

@Override
public void remove() {
 ensureIndexKnown();
 if (curr == -1) throw new IllegalStateException();

 if (curr == prev) {
  /* If the last operation was a next(), we are removing an entry that preceeds
     the current index, and thus we must decrement it. */
  index--;
  prev = GET_PREV(link[curr]);
 }
 else
  next = GET_NEXT(link[curr]);

 size--;
 /* Now we manually fix the pointers. Because of our knowledge of next
    and prev, this is going to be faster than calling fixPointers(). */
 if (prev == -1) first = next;
 else
  SET_NEXT(link[prev], next);
 if (next == -1) last = prev;
 else
  SET_PREV(link[next], prev);

 int last, slot, pos = curr;
 curr = -1;

 if (pos == n) {
  OPEN_HASH_SET.this.containsNull = false;
  OPEN_HASH_SET.this.key[n] = KEY_NULL;
 }
 else
{
  KEY_GENERIC_TYPE curr;
  final KEY_GENERIC_TYPE[] key = OPEN_HASH_SET.this.key;
  // We have to horribly duplicate the shiftKeys() code because we need to update next/prev.
  for(;;) {
   pos = ((last = pos) + 1) & mask;
   for(;;) {
    if (KEY_IS_NULL(curr = key[pos])) {
     key[last] = KEY_NULL;
     return;
    }
    slot = KEY2INTHASH(curr) & mask;
    if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
```

```
      pos = (pos + 1) & mask;
     }

     key[last] = curr;
     if (next == pos) next = last;
     if (prev == pos) prev = last;
     fixPointers(pos, last);
    }
   }
  }
 }


 /** Returns a type-specific list iterator on the elements in this set, starting from a given element of the set.
  * Please see the class documentation for implementation details.
  *
  * @param from an element to start from.
  * @return a type-specific list iterator starting at the given element.
  * @throws IllegalArgumentException if {@code from}
 does not belong to the set.
  */
 @Override
 public KEY_LIST_ITERATOR KEY_GENERIC iterator(KEY_GENERIC_TYPE from) {
  return new SetIterator(from);
 }

 /** Returns a type-specific list iterator on the elements in this set, starting from the first element.
  * Please see the class documentation for implementation details.
  *
  * @return a type-specific list iterator starting at the first element.
  */
 @Override
 public KEY_LIST_ITERATOR KEY_GENERIC iterator() {
  return new SetIterator();
 }

#else

 /** An iterator over a hash set. */

 private class SetIterator implements KEY_ITERATOR KEY_GENERIC {
  /** The index of the last entry returned, if positive or zero; initially, {@link #n}. If negative, the last
   element returned was that of index {@code - pos - 1} from the {@link #wrapped} list. */
  int pos = n;
  /** The index of the last entry that has been returned (more precisely, the value of {@link #pos} if {@link #pos} is
positive,
   or {@link Integer#MIN_VALUE} if {@link
```

```
#pos} is negative). It is -1 if either
 we did not return an entry yet, or the last returned entry has been removed. */
int last = -1;
/** A downward counter measuring how many entries must still be returned. */
int c = size;
/** A boolean telling us whether we should return the null key. */
boolean mustReturnNull = OPEN_HASH_SET.this.containsNull;
/** A lazily allocated list containing elements that have wrapped around the table because of removals. */
ARRAY_LIST KEY_GENERIC wrapped;

@Override
public boolean hasNext() {
 return c != 0;
}

@Override
public KEY_GENERIC_TYPE NEXT_KEY() {
 if (! hasNext()) throw new NoSuchElementException();
 c--;
 if (mustReturnNull) {
  mustReturnNull = false;
  last = n;
  return key[n];
 }
 final KEY_GENERIC_TYPE key[] = OPEN_HASH_SET.this.key;
 for(;;) {
  if (--pos < 0) {
   // We are just enumerating elements from the wrapped list.
   last = Integer.MIN_VALUE;
   return wrapped.GET_KEY(- pos
- 1);
  }
  if (! KEY_IS_NULL(key[pos])) return key[last = pos];
 }
}


/** Shifts left entries with the specified hash code, starting at the specified position,
 * and empties the resulting free entry.
 *
 * @param pos a starting position.
 */
private final void shiftKeys(int pos) {
 // Shift entries with the same hash.
 int last, slot;
 KEY_GENERIC_TYPE curr;
 final KEY_GENERIC_TYPE[] key = OPEN_HASH_SET.this.key;
```

```
  for(;;) {
   pos = ((last = pos) + 1) & mask;

   for(;;) {
    if (KEY_IS_NULL(curr = key[pos])) {
     key[last] = KEY_NULL;
     return;
    }
    slot = KEY2INTHASH(curr) & mask;
    if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
    pos = (pos + 1) & mask;
   }

   if (pos < last) { // Wrapped entry.
    if (wrapped == null) wrapped = new ARRAY_LIST KEY_GENERIC_DIAMOND(2);
    wrapped.add(key[pos]);
   }

   key[last] = curr;
  }
 }

 @Override
 public void remove() {
  if (last == -1)
 throw new IllegalStateException();
  if (last == n) {
   OPEN_HASH_SET.this.containsNull = false;
   OPEN_HASH_SET.this.key[n] = KEY_NULL;
  }
  else if (pos >= 0) shiftKeys(last);
  else {
   // We're removing wrapped entries.
#if KEYS_REFERENCE
   OPEN_HASH_SET.this.remove(wrapped.set(- pos - 1, null));
#else
   OPEN_HASH_SET.this.remove(wrapped.GET_KEY(- pos - 1));
#endif
   last = -1; // Note that we must not decrement size
   return;
  }

  size--;
  last = -1; // You can no longer remove this entry.
  if (ASSERTS) checkTable();
 }
}
```

```
@Override
public KEY_ITERATOR KEY_GENERIC iterator() {
 return new SetIterator();
}

#endif


/** Rehashes this set, making the table as small as possible.
 *
 * <p>This method rehashes the table to the smallest size satisfying the
 * load factor. It can be used when the set will not be changed anymore, so
 * to optimize access speed and size.
 *
 * <p>If the table size is already the minimum possible, this method
 * does
nothing.
 *
 * @return true if there was enough memory to trim the set.
 * @see #trim(int)
 */

public boolean trim() {
 return trim(size);
}


/** Rehashes this set if the table is too large.
 *
 * <p>Let <var>N</var> be the smallest table size that can hold
 * <code>max(n,{@link #size()})</code> entries, still satisfying the load factor. If the current
 * table size is smaller than or equal to <var>N</var>, this method does
 * nothing. Otherwise, it rehashes this set in a table of size
 * <var>N</var>.
 *
 * <p>This method is useful when reusing sets.  {@linkplain #clear() Clearing a
 * set} leaves the table size untouched. If you are reusing a set
 * many times, you can call this method with a typical
 * size to avoid keeping around a very large table just
 * because of a few large transient sets.
 *
 * @param n the threshold for the trimming.
 * @return true if there was enough memory to trim the set.
 * @see #trim()
 */

public boolean trim(final int
```

```
n) {
 final int l = HashCommon.nextPowerOfTwo((int)Math.ceil(n / f));
 if (l >= this.n || size > maxFill(l, f)) return true;
 try {
  rehash(l);
 }
 catch(OutOfMemoryError cantDoIt) { return false; }
 return true;
}

/** Rehashes the set.
 *
 * <p>This method implements the basic rehashing strategy, and may be
 * overriden by subclasses implementing different rehashing strategies (e.g.,
 * disk-based rehashing). However, you should not override this method
 * unless you understand the internal workings of this class.
 *
 * @param newN the new size
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
protected void rehash(final int newN) {
 final KEY_GENERIC_TYPE key[] = this.key;

 final int mask = newN - 1; // Note that this is used by the hashing macro
 final KEY_GENERIC_TYPE newKey[] = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[newN + 1];

#ifdef Linked
 int i = first, prev = -1, newPrev = -1, t, pos;
 final long link[] = this.link;
 final long newLink[] = new long[newN + 1];
 first
= -1;

 for(int j = size; j-- != 0;) {
  if (KEY_EQUALS_NULL(key[i])) pos = newN;
  else {
   pos = KEY2INTHASH(key[i]) & mask;
   while (! KEY_IS_NULL(newKey[pos])) pos = (pos + 1) & mask;
  }

  newKey[pos] = key[i];

  if (prev != -1) {
   SET_NEXT(newLink[newPrev], pos);
   SET_PREV(newLink[pos], newPrev);
   newPrev = pos;
  }
```

```
  else {
   newPrev = first = pos;
   // Special case of SET(newLink[pos], -1, -1);
   newLink[pos] = -1L;
  }

  t = i;
  i = GET_NEXT(link[i]);
  prev = t;
 }

 this.link = newLink;
 this.last = newPrev;
 if (newPrev != -1)
  // Special case of SET_NEXT(newLink[newPrev], -1);
  newLink[newPrev] |= -1 & 0xFFFFFFFFL;
#else
 int i = n, pos;

 for(int j = realSize(); j-- != 0;) {
  while(KEY_IS_NULL(key[--i]));
  if (! KEY_IS_NULL(newKey[pos = KEY2INTHASH(key[i]) & mask]))
   while (! KEY_IS_NULL(newKey[pos = (pos + 1) & mask]));
  newKey[pos] = key[i];
 }
#endif

 n = newN;
 this.mask = mask;
 maxFill = maxFill(n, f);
 this.key
= newKey;
 }


/** Returns a deep copy of this set.
 *
 * <p>This method performs a deep copy of this hash set; the data stored in the
 * set, however, is not cloned. Note that this makes a difference only for object keys.
 *
 *  @return a deep copy of this set.
 */
@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public OPEN_HASH_SET KEY_GENERIC clone() {
 OPEN_HASH_SET KEY_GENERIC c;
 try {
  c = (OPEN_HASH_SET KEY_GENERIC)super.clone();
```

```
  }
  catch(CloneNotSupportedException cantHappen) {
   throw new InternalError();
  }
  c.key = key.clone();
  c.containsNull = containsNull;
#ifdef Linked
  c.link = link.clone();
#endif
#ifdef Custom
  c.strategy = strategy;
#endif
  return c;
 }

 /** Returns a hash code for this set.
  *
  * This method overrides the generic method provided by the superclass.
  * Since {@code equals()} is not overriden, it is important
  * that the value returned by this method is the same value as
  * the one returned by the overriden method.

 *
  * @return a hash code for this set.
  */
 @Override
 public int hashCode() {
  int h = 0;
  for(int j = realSize(), i = 0; j-- != 0;) {
   while(KEY_IS_NULL(key[i])) i++;
#if KEYS_REFERENCE
   if (this != key[i])
#endif
    h += KEY2JAVAHASH_NOT_NULL(key[i]);
   i++;
  }

  // Zero / null have hash zero.
  return h;
 }


 private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
  final KEY_ITERATOR KEY_GENERIC i = iterator();
  s.defaultWriteObject();
  for(int j = size; j-- != 0;) s.WRITE_KEY(i.NEXT_KEY());
 }
```

```
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
  s.defaultReadObject();

  n = arraySize(size, f);
  maxFill = maxFill(n, f);
  mask = n - 1;

  final KEY_GENERIC_TYPE key[] = this.key = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[n + 1];
#ifdef Linked
  final long link[] = this.link = new long[n + 1];
  int prev = -1;
  first = last = -1;
#endif

  KEY_GENERIC_TYPE
 k;

  for(int i = size, pos; i-- != 0;) {

   k = KEY_GENERIC_CAST s.READ_KEY();

   if (KEY_EQUALS_NULL(k)) {
    pos = n;
    containsNull = true;
   }
   else {
    if (! KEY_IS_NULL(key[pos = KEY2INTHASH(k) & mask]))
     while (! KEY_IS_NULL(key[pos = (pos + 1) & mask]));
   }

   key[pos] = k;

#ifdef Linked
   if (first != -1) {
    SET_NEXT(link[prev], pos);
    SET_PREV(link[pos], prev);
    prev = pos;
   }
   else {
    prev = first = pos;
    // Special case of SET_PREV(newLink[pos], -1);
    link[pos] |= (-1L & 0xFFFFFFFFL) << 32;
   }
#endif
  }
```

```
#ifdef Linked
  last = prev;
  if (prev != -1)
   // Special case of SET_NEXT(link[prev], -1);
   link[prev] |= -1 & 0xFFFFFFFFL;
#endif

  if (ASSERTS) checkTable();
 }


#ifdef ASSERTS_CODE
 private void checkTable() {
  assert (n & -n) == n : "Table length is not a power of two: " + n;
  assert n == key.length - 1;
  int n = key.length - 1;
  while(n-- != 0)
   if (! KEY_IS_NULL(key[n]) && ! contains(key[n]))
    throw new
 AssertionError("Hash table has key " + key[n] + " marked as occupied, but the key does not belong to the table");

#if KEYS_PRIMITIVE
  java.util.HashSet<KEY_GENERIC_CLASS> s = new java.util.HashSet<KEY_GENERIC_CLASS> ();
#else
  java.util.HashSet<Object> s = new java.util.HashSet<Object>();
#endif

  for(int i = key.length - 1; i-- != 0;)
   if (! KEY_IS_NULL(key[i]) && ! s.add(key[i])) throw new AssertionError("Key " + key[i] + " appears twice at
position " + i);

#ifdef Linked
  KEY_LIST_ITERATOR KEY_GENERIC i = iterator();
  KEY_GENERIC_TYPE k;
  n = size();
  while(n-- != 0)
   if (! contains(k = i.NEXT_KEY()))
    throw new AssertionError("Linked hash table forward enumerates key " + k + ", but the key does not belong to the
table");

  if (i.hasNext()) throw new AssertionError("Forward iterator not exhausted");

  n = size();
  if (n > 0) {
   i = iterator(LAST());
   while(n-- != 0)
    if (! contains(k = i.PREV_KEY()))
     throw new AssertionError("Linked hash table backward
```

```
 enumerates key " + k + ", but the key does not belong to the table");

   if (i.hasPrevious()) throw new AssertionError("Previous iterator not exhausted");
  }
#endif
 }
#else
 private void checkTable() {}
#endif

#ifdef TEST

 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif KEY_CLASS_Object
#ifdef Custom
  int i = r.nextInt(3);
  byte a[] = new byte[i];
  while(i-- != 0) a[i] = (byte)r.nextInt();
  return a;
#else
  return Integer.toBinaryString(r.nextInt());
#endif
#else
  return new java.io.Serializable() {};
#endif
 }


 private static final class ArrayComparator implements java.util.Comparator {
  public int compare(Object a, Object b) {
   byte[] aa = (byte[])a;
   byte[] bb = (byte[])b;
   int length = Math.min(aa.length, bb.length);
   for(int
 i = 0; i < length; i++) {
    if (aa[i] < bb[i]) return -1;
    if (aa[i] > bb[i]) return 1;
   }
   return aa.length == bb.length ? 0 : (aa.length < bb.length ? -1 : 1);
  }
 }
```

```
private static final class MockSet extends java.util.TreeSet {
 private java.util.List list = new java.util.ArrayList();

 public MockSet(java.util.Comparator c) { super(c); }

 public boolean add(Object k) {
  if (! contains(k)) list.add(k);
  return super.add(k);
 }

 public boolean addAll(Collection c) {
  java.util.Iterator i = c.iterator();
  boolean result = false;
  while(i.hasNext()) result |= add(i.next());
  return result;
 }

 public boolean removeAll(Collection c) {
  java.util.Iterator i = c.iterator();
  boolean result = false;
  while(i.hasNext()) result |= remove(i.next());
  return result;
 }

 public boolean remove(Object k) {
  if (contains(k)) {
   int i = list.size();
   while(i-- != 0) if (comparator().compare(list.get(i), k) == 0) {
    list.remove(i);
    break;
   }
  }
  return
super.remove(k);
 }

 private void justRemove(Object k) { super.remove(k); }

 public java.util.Iterator iterator() {
  return new java.util.Iterator() {
    final java.util.Iterator iterator = list.iterator();
    Object curr;
    public Object next() { return curr = iterator.next(); }
    public boolean hasNext() { return iterator.hasNext(); }
    public void remove() {
     justRemove(curr);
     iterator.remove();
```

```
    }
   };
  }
 }

 private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
 private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

 private static String format(double d) {
  StringBuffer s = new StringBuffer();
  return format.format(d, s, fp).toString();
 }

 private static void speedTest(int n, float f, boolean comp) {
#ifndef Custom
  int i, j;
  OPEN_HASH_SET m;
#ifdef Linked
  java.util.LinkedHashSet t;
#else
  java.util.HashSet t;
#endif

  KEY_TYPE k[] = new
 KEY_TYPE[n];
  KEY_TYPE nk[] = new KEY_TYPE[n];
  long ns;

  for(i = 0; i < n; i++) {
   k[i] = genKey();
   nk[i] = genKey();
  }

  double totAdd = 0, totYes = 0, totNo = 0, totIter = 0, totRemYes = 0, totRemNo = 0, d;

  if (comp) { for(j = 0; j < 20; j++) {

#ifdef Linked
   t = new java.util.LinkedHashSet(16);
#else
   t = new java.util.HashSet(16);
#endif

   /* We add pairs to t. */
   ns = System.nanoTime();
   for(i = 0; i < n;  i++) t.add(KEY2OBJ(k[i]));
   d = (System.nanoTime() - ns) / (double)n;
   if (j > 2) totAdd += d;
```

```
  System.out.print("Add: " + format(d) + "ns ");

  /* We check for pairs in t. */
  ns = System.nanoTime();
  for(i = 0; i < n;  i++) t.contains(KEY2OBJ(k[i]));
  d = (System.nanoTime() - ns) / (double)n;
  if (j > 2) totYes += d;
  System.out.print("Yes: " + format(d) + "ns ");

  /* We check for pairs not in t. */
  ns = System.nanoTime();
  for(i = 0; i < n;  i++) t.contains(KEY2OBJ(nk[i]));
  d = (System.nanoTime() - ns) / (double)n;
  if
 (j > 2) totNo += d;
  System.out.print("No: " + format(d) + "ns ");

  /* We iterate on t. */
  ns = System.nanoTime();
  for(java.util.Iterator it = t.iterator(); it.hasNext(); it.next());
  d = (System.nanoTime() - ns) / (double)n;
  if (j > 2) totIter += d;
  System.out.print("Iter: " + format(d) + "ns ");

  // Too expensive in the linked case
#ifndef Linked
  /* We delete pairs not in t. */
  ns = System.nanoTime();
  for(i = 0; i < n;  i++) t.remove(KEY2OBJ(nk[i]));
  d = (System.nanoTime() - ns) / (double)n;
  if (j > 2) totRemNo += d;
  System.out.print("RemNo: " + format(d) + "ns ");

  /* We delete pairs in t. */
  ns = System.nanoTime();
  for(i = 0; i < n;  i++) t.remove(KEY2OBJ(k[i]));
  d = (System.nanoTime() - ns) / (double)n;
  if (j > 2) totRemYes += d;
  System.out.print("RemYes: " + format(d) + "ns ");
#endif

  System.out.println();
  }

  System.out.println();
  System.out.println("java.util Add: " + format(totAdd/(j-3)) + "ns Yes: " + format(totYes/(j-3))
 + "ns No: " + format(totNo/(j-3)) + "ns Iter: " + format(totIter/(j-3)) + "ns RemNo: " + format(totRemNo/(j-3)) +
"ns RemYes: " + format(totRemYes/(j-3)) + "ns");
```

```
  System.out.println();

  totAdd = totYes = totNo = totIter = totRemYes = totRemNo = 0;
  }

  for(j = 0; j < 20; j++) {

   m = new OPEN_HASH_SET(16, f);

   /* We add pairs to m. */
   ns = System.nanoTime();
   for(i = 0; i < n;  i++) m.add(k[i]);
   d = (System.nanoTime() - ns) / (double)n;
   if (j > 2) totAdd += d;
   System.out.print("Add: " + format(d) + "ns ");

   /* We check for pairs in m. */
   ns = System.nanoTime();
   for(i = 0; i < n;  i++) m.contains(k[i]);
   d = (System.nanoTime() - ns) / (double)n;
   if (j > 2) totYes += d;
   System.out.print("Yes: " + format(d) + "ns ");

   /* We check for pairs not in m. */
   ns = System.nanoTime();
   for(i = 0; i < n;  i++) m.contains(nk[i]);
   d = (System.nanoTime() - ns) / (double)n;
   if (j > 2) totNo += d;
   System.out.print("No: " + format(d)
 + "ns ");

   /* We iterate on m. */
   ns = System.nanoTime();
   for(KEY_ITERATOR it = (KEY_ITERATOR)m.iterator(); it.hasNext(); it.NEXT_KEY());
   d = (System.nanoTime() - ns) / (double)n;
   if (j > 2) totIter += d;
   System.out.print("Iter: " + format(d) + "ns ");

   // Too expensive in the linked case
#ifndef Linked
   /* We delete pairs not in m. */
   ns = System.nanoTime();
   for(i = 0; i < n;  i++) m.remove(nk[i]);
   d = (System.nanoTime() - ns) / (double)n;
   if (j > 2) totRemNo += d;
   System.out.print("RemNo: " + format(d) + "ns ");
```

```
  /* We delete pairs in m. */
  ns = System.nanoTime();
  for(i = 0; i < n;  i++) m.remove(k[i]);
  d = (System.nanoTime() - ns) / (double)n;
  if (j > 2) totRemYes += d;
  System.out.print("RemYes: " + format(d) + "ns ");
#endif

  System.out.println();
  }


  System.out.println();
  System.out.println("fastutil  Add: " + format(totAdd/(j-3)) + "ns Yes: " + format(totYes/(j-3)) + "ns No: " +
format(totNo/(j-3)) + "ns Iter:
 " + format(totIter/(j-3)) + "ns RemNo: " + format(totRemNo/(j-3)) + "ns RemYes: " + format(totRemYes/(j-3)) +
"ns");

  System.out.println();
#endif
 }


 private static void fatal(String msg) {
  System.out.println(msg);
  System.exit(1);
 }

 private static void ensure(boolean cond, String msg) {
  if (cond) return;
  fatal(msg);
 }


 private static void printProbes(OPEN_HASH_SET m) {
  long totProbes = 0;
  double totSquareProbes = 0;
  int maxProbes = 0;
  final double f = (double)m.size / m.n;
  for(int i = 0, c = 0; i < m.n; i++) {
   if (! KEY_IS_NULL(m.key[i])) c++;
   else {
    if (c != 0) {
     final long p = (c + 1) * (c + 2) / 2;
     totProbes += p;
     totSquareProbes += (double)p * p;
    }
    maxProbes = Math.max(c, maxProbes);
```

```
    c = 0;
    totProbes++;
    totSquareProbes++;
   }
  }

  final double expected = (double)totProbes / m.n;
  System.err.println("Expected probes: " + (
   3 * Math.sqrt(3) * (f / ((1 - f) * (1 - f))) + 4 / (9 * f) - 1
  ) + "; actual: "
 + expected + "; stddev: " + Math.sqrt(totSquareProbes / m.n - expected * expected)  + "; max probes: " +
maxProbes);
 }


 private static void runTest(int n, float f) {
#if !defined(Custom) || KEYS_REFERENCE

  int c;
#ifdef Custom
  OPEN_HASH_SET m = new OPEN_HASH_SET(Hash.DEFAULT_INITIAL_SIZE, f,
it.unimi.dsi.fastutil.bytes.ByteArrays.HASH_STRATEGY);
#else
  OPEN_HASH_SET m = new OPEN_HASH_SET(Hash.DEFAULT_INITIAL_SIZE, f);
#endif
#ifdef Linked
#ifdef Custom
  java.util.Set t = new MockSet(new ArrayComparator());
#else
  java.util.Set t = new java.util.LinkedHashSet();
#endif
#else
#ifdef Custom
  java.util.Set t = new java.util.TreeSet(new ArrayComparator());
#else
  java.util.Set t = new java.util.HashSet();
#endif
#endif

  /* First of all, we fill t with random data. */

  for(int i=0; i<f * n;  i++) t.add(KEY2OBJ(genKey()));

  /* Now we add to m the same data */

  m.addAll(t);

  if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after insertion");
```

```
 if
(!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after insertion");
 printProbes(m);

 /* Now we check that m actually holds that data. */

 for(java.util.Iterator i=t.iterator(); i.hasNext();) {
  Object e = i.next();
  if (!m.contains(e)) {
   System.out.println("Error (" + seed + "): m and t differ on a key ("+e+") after insertion (iterating on t)");
   System.exit(1);
  }
 }

 /* Now we check that m actually holds that data, but iterating on m. */

 c = 0;
 for(java.util.Iterator i=m.iterator(); i.hasNext();) {
  Object e = i.next();
  c++;
  if (!t.contains(e)) {
   System.out.println("Error (" + seed + "): m and t differ on a key ("+e+") after insertion (iterating on m)");
   System.exit(1);
  }
 }

 if (c != t.size()) {
  System.out.println("Error (" + seed + "): m has only " + c + " keys instead of " + t.size() + " after insertion (iterating
on m)");
  System.exit(1);
 }
 /* Now we check that inquiries about random data give the same
 answer in m and t. For
    m we use the polymorphic method. */

 for(int i=0; i<n;  i++) {
  KEY_TYPE T = genKey();
  if (m.contains(T) != t.contains(KEY2OBJ(T))) {
   System.out.println("Error (" + seed + "): divergence in keys between t and m (polymorphic method)");
   System.exit(1);
  }
 }

 /* Again, we check that inquiries about random data give the same answer in m and t, but
    for m we use the standard method. */

 for(int i=0; i<n;  i++) {
  KEY_TYPE T = genKey();
```

```
 if (m.contains(KEY2OBJ(T)) != t.contains(KEY2OBJ(T))) {
  System.out.println("Error (" + seed + "): divergence between t and m (standard method)");
  System.exit(1);
 }
}


/* Now we put and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<20*n;  i++) {
 KEY_TYPE T = genKey();
 if (m.add(KEY2OBJ(T)) != t.add(KEY2OBJ(T))) {
  System.out.println("Error (" + seed + "): divergence in add() between t and m");
  System.exit(1);
 }
 T = genKey();
 if
(m.remove(KEY2OBJ(T)) != t.remove(KEY2OBJ(T))) {
  System.out.println("Error (" + seed + "): divergence in remove() between t and m");
  System.exit(1);
 }
}

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after removal");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after removal");

/* Now we check that m actually holds that data. */

for(java.util.Iterator i=t.iterator(); i.hasNext();) {
 Object e = i.next();
 if (!m.contains(e)) {
  System.out.println("Error (" + seed + "): m and t differ on a key ("+e+") after removal (iterating on t)");
  System.exit(1);
 }
}

/* Now we check that m actually holds that data, but iterating on m. */

for(java.util.Iterator i=m.iterator(); i.hasNext();) {
 Object e = i.next();
 if (!t.contains(e)) {
  System.out.println("Error (" + seed + "): m and t differ on a key ("+e+") after removal (iterating on m)");
  System.exit(1);
 }
}

printProbes(m);
```

```
  /*
 Now we make m into an array, make it again a set and check it is OK. */
  KEY_TYPE a[] = m.TO_KEY_ARRAY();

#ifdef Custom
  if (!new OPEN_HASH_SET(a, m.strategy()).equals(m))
   System.out.println("Error (" + seed + "): toArray() output (or array-based constructor) is not OK");
#else
  if (!new OPEN_HASH_SET(a).equals(m))
   System.out.println("Error (" + seed + "): toArray() output (or array-based constructor) is not OK");
#endif

  /* Now we check cloning. */

  ensure(m.equals(((OPEN_HASH_SET)m).clone()), "Error (" + seed + "): m does not equal m.clone()");
  ensure(((OPEN_HASH_SET)m).clone().equals(m), "Error (" + seed + "): m.clone() does not equal m");

  int h = m.hashCode();

  /* Now we save and read m. */

  try {
   java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
   java.io.OutputStream os = new java.io.FileOutputStream(ff);
   java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

   oos.writeObject(m);
   oos.close();

   java.io.InputStream
 is = new java.io.FileInputStream(ff);
   java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

   m = (OPEN_HASH_SET)ois.readObject();
   ois.close();
   ff.delete();
  }
  catch(Exception e) {
   e.printStackTrace();
   System.exit(1);
  }

#if !KEY_CLASS_Reference
  if (m.hashCode() != h) System.out.println("Error (" + seed + "): hashCode() changed after save/read");

  printProbes(m);

  /* Now we check that m actually holds that data, but iterating on m. */
```

```
  for(java.util.Iterator i=m.iterator(); i.hasNext();) {
   Object e = i.next();
   if (!t.contains(e)) {
    System.out.println("Error (" + seed + "): m and t differ on a key ("+e+") after save/read");
    System.exit(1);
   }
  }
#else
  m.clear();
  m.addAll(t);
#endif

  /* Now we put and remove random data in m and t, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();
   if (m.add(KEY2OBJ(T)) != t.add(KEY2OBJ(T)))
  {
    System.out.println("Error (" + seed + "): divergence in add() between t and m after save/read");
    System.exit(1);
   }
   T = genKey();
   if (m.remove(KEY2OBJ(T)) != t.remove(KEY2OBJ(T))) {
    System.out.println("Error (" + seed + "): divergence in remove() between t and m after save/read");
    System.exit(1);
   }
  }

  if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after post-save/read removal");
  if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after post-save/read removal");


#ifdef Linked


  /* Now we play with iterators, but only in the linked case. */

  {
   java.util.ListIterator i, j;
   Object I, J;
   i = (java.util.ListIterator)m.iterator();
   j = new java.util.LinkedList(t).listIterator();

   for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + seed + "): divergence in hasNext()");
    ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + seed + "): divergence in hasPrevious()");
```

```
    if
 (r.nextFloat() < .8 && i.hasNext()) {
#ifdef Custom
     ensure(m.strategy().equals(i.next(), J = j.next()), "Error (" + seed + "): divergence in next()");
#else
     ensure(i.next().equals(J = j.next()), "Error (" + seed + "): divergence in next()");
#endif
     if (r.nextFloat() < 0.5) {
      i.remove();
      j.remove();
      t.remove(J);
     }
    }
    else if (r.nextFloat() < .2 && i.hasPrevious()) {
#ifdef Custom
     ensure(m.strategy().equals(i.previous(), J = j.previous()), "Error (" + seed + "): divergence in previous()");
#else
     ensure(i.previous().equals(J = j.previous()), "Error (" + seed + "): divergence in previous()");
#endif
     if (r.nextFloat() < 0.5) {
      i.remove();
      j.remove();
      t.remove(J);
     }
    }

    ensure(i.nextIndex() == j.nextIndex(), "Error (" + seed + "): divergence in nextIndex()");
    ensure(i.previousIndex() == j.previousIndex(), "Error (" + seed + "): divergence in previousIndex()");

   }

  }

  if (t.size()
 > 0) {
   java.util.ListIterator i, j;
   Object J;
   j = new java.util.LinkedList(t).listIterator();
   int e = r.nextInt(t.size());
   Object from;
   do from = j.next(); while(e-- != 0);

   i = (java.util.ListIterator)m.iterator(KEY_OBJ2TYPE(from));

   for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + seed + "): divergence in hasNext() (iterator with starting point " +
from + ")");
    ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + seed + "): divergence in hasPrevious() (iterator with starting
```

```
point " + from + ")");

    if (r.nextFloat() < .8 && i.hasNext()) {
#ifdef Custom
     ensure(m.strategy().equals(i.next(), J = j.next()), "Error (" + seed + "): divergence in next() (iterator with starting
point " + from + ")");
#else
     ensure(i.next().equals(J = j.next()), "Error (" + seed + "): divergence in next() (iterator with starting point " + from +
")");
#endif

     if (r.nextFloat() < 0.5) {
      i.remove();
      j.remove();
      t.remove(J);
     }
    }
    else
 if (r.nextFloat() < .2 && i.hasPrevious()) {
#ifdef Custom
     ensure(m.strategy().equals(i.previous(), J = j.previous()), "Error (" + seed + "): divergence in previous() (iterator
with starting point " + from + ")");
#else
     ensure(i.previous().equals(J = j.previous()), "Error (" + seed + "): divergence in previous() (iterator with starting
point " + from + ")");
#endif

     if (r.nextFloat() < 0.5) {
      i.remove();
      j.remove();
      t.remove(J);
     }
    }

    ensure(i.nextIndex() == j.nextIndex(), "Error (" + seed + "): divergence in nextIndex() (iterator with starting point "
+ from + ")");
    ensure(i.previousIndex() == j.previousIndex(), "Error (" + seed + "): divergence in previousIndex() (iterator with
starting point " + from + ")");

   }

  }

  /* Now we check that m actually holds that data. */

  ensure(m.equals(t), "Error (" + seed + "): ! m.equals(t) after iteration");
  ensure(t.equals(m), "Error (" + seed + "):
```

```
 ! t.equals(m) after iteration");



#endif

  /* Now we take out of m everything, and check that it is empty. */

  for(java.util.Iterator i=m.iterator(); i.hasNext();) { i.next(); i.remove();}

  if (!m.isEmpty())  {
   System.out.println("Error (" + seed + "): m is not empty (as it should be)");
   System.exit(1);
  }

#if KEY_CLASS_Integer || KEY_CLASS_Long
  m = new OPEN_HASH_SET(n, f);
  t.clear();
  int x;

  /* Now we torture-test the hash table. This part is implemented only for integers and longs. */

  int p = m.key.length - 1;

  for(int i=0; i<p; i++) {
   for (int j=0; j<20; j++) {
    m.add(i+(r.nextInt() % 10)*p);
    m.remove(i+(r.nextInt() % 10)*p);
   }

   for (int j=-10; j<10; j++) m.remove(i+j*p);
  }

  t.addAll(m);

  /* Now all table entries are REMOVED. */

  int k = 0;
  for(int i=0; i<(p*f)/10; i++) {
   for (int j=0; j<10; j++) {
    k++;
    x = i+(r.nextInt() % 10)*p;
    if (m.add(x) != t.add(KEY2OBJ(x)))
     System.out.println("Error (" + seed + "): m
 and t differ on a key during torture-test insertion.");
   }
  }
```

```
  if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after torture-test insertion");
  if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after torture-test insertion");

  for(int i=0; i<(p*f)/10; i++) {
   for (int j=0; j<10; j++) {
    x = i+(r.nextInt() % 10)*p;
    if (m.remove(x) != t.remove(KEY2OBJ(x)))
     System.out.println("Error (" + seed + "): m and t differ on a key during torture-test removal.");
   }
  }

  if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after torture-test removal");
  if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after torture-test removal");

  if (!m.equals(m.clone())) System.out.println("Error (" + seed + "): !m.equals(m.clone()) after torture-test removal");
  if (!((OPEN_HASH_SET)m.clone()).equals(m)) System.out.println("Error (" + seed + "): !m.clone().equals(m) after
torture-test removal");

  m.trim();

  if
 (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after trim()");
  if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after trim()");
#endif

  System.out.println("Test OK");
  return;
#endif
 }


 public static void main(String args[]) {
  float f = Hash.DEFAULT_LOAD_FACTOR;
  int n  = Integer.parseInt(args[1]);
  if (args.length>2) f = Float.parseFloat(args[2]);
  if (args.length > 3) r = new java.util.Random(seed = Long.parseLong(args[3]));

  try {
   if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, f, "speedComp".equals(args[0]));
   else if ("test".equals(args[0])) runTest(n, f);
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed: " + seed);
  }
 }

#endif
```

```
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/OpenHashSet.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

/** An abstract class providing basic methods for functions implementing a type-specific interface.
*
* <p>This class handles directly a default return
* value (including {@linkplain #defaultReturnValue() methods to access
* it}). Instances of classes inheriting from this class have just to return
 * {@code defRetValue} to denote lack of a key in type-specific methods. The value
* is serialized.
*
* <p>Implementing subclasses have just to provide type-specific {@code get()},
* type-specific {@code containsKey()}, and {@code size()} methods.
*
*/

public abstract class ABSTRACT_FUNCTION KEY_VALUE_GENERIC implements FUNCTION
KEY_VALUE_GENERIC, java.io.Serializable {

 private static final long serialVersionUID = -4940583368468432370L;

 protected ABSTRACT_FUNCTION() {}

 /**
  * The default return value for {@code get()}, {@code put()} and
```

```
 * {@code remove()}.
 */

 protected VALUE_GENERIC_TYPE defRetValue;

 @Override
 public void defaultReturnValue(final VALUE_GENERIC_TYPE rv) {
  defRetValue = rv;
 }

 @Override
 public VALUE_GENERIC_TYPE defaultReturnValue() {
  return defRetValue;
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractFunction.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.SortedSet;
import java.util.NoSuchElementException;
#if KEYS_REFERENCE
import java.util.Comparator;
#endif


/** A class providing static methods and objects that do useful things with type-specific sorted sets.
```

```
*
* @see java.util.Collections
*/

public final class SORTED_SETS {

 private
 SORTED_SETS() {}

 /** An immutable class representing the empty sorted set and implementing a type-specific set interface.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific sorted set.
  */

 public static class EmptySet KEY_GENERIC extends SETS.EmptySet KEY_GENERIC implements
SORTED_SET KEY_GENERIC, java.io.Serializable, Cloneable {
  private static final long serialVersionUID = -7046029254386353129L;

  protected EmptySet() {}

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
  public KEY_BIDI_ITERATOR KEY_GENERIC iterator(KEY_GENERIC_TYPE from) { return
ITERATORS.EMPTY_ITERATOR; }

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
  public SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE to) { return
EMPTY_SET; }

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
  public SORTED_SET KEY_GENERIC headSet(KEY_GENERIC_TYPE from) { return EMPTY_SET; }

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
  public SORTED_SET KEY_GENERIC
 tailSet(KEY_GENERIC_TYPE to) { return EMPTY_SET; }

  @Override
  public KEY_GENERIC_TYPE FIRST() { throw new NoSuchElementException(); }

  @Override
  public KEY_GENERIC_TYPE LAST() { throw new NoSuchElementException(); }

  @Override
  public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return null; }
```

```
#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_CLASS from, KEY_GENERIC_CLASS to) {
return EMPTY_SET; }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_SET KEY_GENERIC headSet(KEY_GENERIC_CLASS from) { return EMPTY_SET; }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_SET KEY_GENERIC tailSet(KEY_GENERIC_CLASS to) { return EMPTY_SET; }

  /**
 {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS first() { throw new NoSuchElementException(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_GENERIC_CLASS last() { throw new NoSuchElementException(); }
#endif

  @Override
  public Object clone() { return EMPTY_SET; }

  private Object readResolve() { return EMPTY_SET; }
 }


 /** An empty sorted set (immutable). It is serializable and cloneable.
  *
  */
 SUPPRESS_WARNINGS_KEY_RAWTYPES
 public static final EmptySet EMPTY_SET = new EmptySet();
```

```
#if KEYS_REFERENCE
 /** Returns an empty sorted set (immutable). It is serializable and cloneable.
  *
  * <p>This method provides a typesafe access to {@link #EMPTY_SET}.
  * @return an empty sorted set (immutable).
  */
 @SuppressWarnings("unchecked")
 public static KEY_GENERIC SET KEY_GENERIC emptySet() {
  return
 EMPTY_SET;
 }
#endif

 /** A class representing a singleton sorted set.
  *
  * <p>This class may be useful to implement your own in case you subclass
  * a type-specific sorted set.
  */

 public static class Singleton KEY_GENERIC extends SETS.Singleton KEY_GENERIC implements SORTED_SET
KEY_GENERIC, java.io.Serializable, Cloneable {

  private static final long serialVersionUID = -7046029254386353129L;

  final KEY_COMPARATOR KEY_SUPER_GENERIC comparator;

  protected Singleton(final KEY_GENERIC_TYPE element, final KEY_COMPARATOR KEY_SUPER_GENERIC
comparator) {
   super(element);
   this.comparator = comparator;
  }

  private Singleton(final KEY_GENERIC_TYPE element) {
   this(element, null);
  }

  SUPPRESS_WARNINGS_KEY_UNCHECKED
  final int compare(final KEY_GENERIC_TYPE k1, final KEY_GENERIC_TYPE k2) {
   return comparator == null ? KEY_CMP(k1, k2) : comparator.compare(k1, k2);
  }

  @Override
  public KEY_BIDI_ITERATOR KEY_GENERIC iterator(KEY_GENERIC_TYPE from) {
   KEY_BIDI_ITERATOR
 KEY_GENERIC i = iterator();
   if (compare(element, from) <= 0) i.NEXT_KEY();
   return i;
  }
```

```
  @Override
  public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return comparator; }

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
  public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_TYPE from, final KEY_GENERIC_TYPE
to) { if (compare(from, element) <= 0 && compare(element, to) < 0) return this; return EMPTY_SET; }

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
  public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) { if (compare(element, to) < 0)
return this; return EMPTY_SET; }

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
  public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) { if (compare(from, element)
<= 0) return this; return EMPTY_SET; }

  @Override
  public KEY_GENERIC_TYPE FIRST() { return element; }

  @Override
  public KEY_GENERIC_TYPE LAST() { return element; }

#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please
 use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_SET KEY_GENERIC subSet(final KEY_CLASS from, final KEY_CLASS to) { return
subSet(KEY_CLASS2TYPE(from), KEY_CLASS2TYPE(to)); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_SET KEY_GENERIC headSet(final KEY_CLASS to) { return headSet(KEY_CLASS2TYPE(to));
}

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_SET KEY_GENERIC tailSet(final KEY_CLASS from) { return
tailSet(KEY_CLASS2TYPE(from)); }
```

```
 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 public KEY_CLASS first() { return KEY2OBJ(element); }

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 public KEY_CLASS
last() { return KEY2OBJ(element); }
#endif
}


/** Returns a type-specific immutable sorted set containing only the specified element. The returned sorted set is
serializable and cloneable.
 *
 * @param element the only element of the returned sorted set.
 * @return a type-specific immutable sorted set containing just {@code element}.
 */

public static KEY_GENERIC SORTED_SET KEY_GENERIC singleton(final KEY_GENERIC_TYPE element) {
 return new Singleton KEY_GENERIC_DIAMOND(element);
}

/** Returns a type-specific immutable sorted set containing only the specified element, and using a specified
comparator. The returned sorted set is serializable and cloneable.
 *
 * @param element the only element of the returned sorted set.
 * @param comparator the comparator to use in the returned sorted set.
 * @return a type-specific immutable sorted set containing just {@code element}.
 */

public static KEY_GENERIC SORTED_SET KEY_GENERIC singleton(final KEY_GENERIC_TYPE element,
final KEY_COMPARATOR KEY_SUPER_GENERIC comparator) {
 return new Singleton KEY_GENERIC_DIAMOND(element, comparator);
}

#if KEYS_PRIMITIVE

/** Returns a type-specific immutable sorted set containing only the specified element. The returned sorted set is
serializable and cloneable.
 *
 * @param element the only element of the returned sorted set.
 * @return a type-specific immutable sorted set containing just {@code element}.
 */
```

```
 public static KEY_GENERIC SORTED_SET KEY_GENERIC singleton(final Object element) {
  return new Singleton(KEY_OBJ2TYPE(element));
 }

 /** Returns a type-specific immutable sorted set containing only the specified element, and using a specified
comparator. The returned sorted set is serializable and cloneable.
  *
  * @param element the only element of the returned sorted set.
  * @param comparator the comparator to use in the returned sorted set.
  * @return a type-specific immutable sorted set containing just {@code element}.
  */

 public static
 KEY_GENERIC SORTED_SET KEY_GENERIC singleton(final Object element, final KEY_COMPARATOR
KEY_SUPER_GENERIC comparator) {
  return new Singleton(KEY_OBJ2TYPE(element), comparator);
 }
#endif


 /** A synchronized wrapper class for sorted sets. */

 public static class SynchronizedSortedSet KEY_GENERIC extends SETS.SynchronizedSet KEY_GENERIC
implements SORTED_SET KEY_GENERIC, java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final SORTED_SET KEY_GENERIC sortedSet;

  protected SynchronizedSortedSet(final SORTED_SET KEY_GENERIC s, final Object sync) {
   super(s, sync);
   sortedSet = s;
  }

  protected SynchronizedSortedSet(final SORTED_SET KEY_GENERIC s) {
   super(s);
   sortedSet = s;
  }

  @Override
  public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { synchronized(sync) { return
sortedSet.comparator(); } }

  @Override
  public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_TYPE from, final KEY_GENERIC_TYPE
to) { return
```

```
 new SynchronizedSortedSet KEY_GENERIC_DIAMOND(sortedSet.subSet(from, to), sync); }

  @Override
  public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) { return new
SynchronizedSortedSet KEY_GENERIC_DIAMOND(sortedSet.headSet(to), sync); }

  @Override
  public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) { return new
SynchronizedSortedSet KEY_GENERIC_DIAMOND(sortedSet.tailSet(from), sync); }

  @Override
  public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return sortedSet.iterator(); }

  @Override
  public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return
sortedSet.iterator(from); }

  @Override
  public KEY_GENERIC_TYPE FIRST() {  synchronized(sync) { return sortedSet.FIRST(); } }

  @Override
  public KEY_GENERIC_TYPE LAST() {  synchronized(sync) { return sortedSet.LAST(); } }

#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public
 KEY_CLASS first() {  synchronized(sync) { return sortedSet.first(); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_CLASS last() {  synchronized(sync) { return sortedSet.last(); } }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_SET KEY_GENERIC subSet(final KEY_CLASS from, final KEY_CLASS to) { return new
SynchronizedSortedSet(sortedSet.subSet(from, to), sync); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
```

```
  public SORTED_SET KEY_GENERIC headSet(final KEY_CLASS to) { return new
SynchronizedSortedSet(sortedSet.headSet(to), sync); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_SET KEY_GENERIC tailSet(final KEY_CLASS from) { return
 new SynchronizedSortedSet(sortedSet.tailSet(from), sync); }
#endif
 }


 /** Returns a synchronized type-specific sorted set backed by the given type-specific sorted set.
  *
  * @param s the sorted set to be wrapped in a synchronized sorted set.
  * @return a synchronized view of the specified sorted set.
  * @see java.util.Collections#synchronizedSortedSet(SortedSet)
  */
 public static KEY_GENERIC SORTED_SET KEY_GENERIC synchronize(final SORTED_SET KEY_GENERIC
s) { return new SynchronizedSortedSet KEY_GENERIC_DIAMOND(s); }

 /** Returns a synchronized type-specific sorted set backed by the given type-specific sorted set, using an assigned
object to synchronize.
  *
  * @param s the sorted set to be wrapped in a synchronized sorted set.
  * @param sync an object that will be used to synchronize the access to the sorted set.
  * @return a synchronized view of the specified sorted set.
  * @see java.util.Collections#synchronizedSortedSet(SortedSet)
  */

 public static KEY_GENERIC
 SORTED_SET KEY_GENERIC synchronize(final SORTED_SET KEY_GENERIC s, final Object sync) { return
new SynchronizedSortedSet KEY_GENERIC_DIAMOND(s, sync); }




 /** An unmodifiable wrapper class for sorted sets. */

 public static class UnmodifiableSortedSet KEY_GENERIC extends SETS.UnmodifiableSet KEY_GENERIC
implements SORTED_SET KEY_GENERIC, java.io.Serializable {

  private static final long serialVersionUID = -7046029254386353129L;

  protected final SORTED_SET KEY_GENERIC sortedSet;
```

```
  protected UnmodifiableSortedSet(final SORTED_SET KEY_GENERIC s) {
   super(s);
   sortedSet = s;
  }

  @Override
  public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return sortedSet.comparator(); }

  @Override
  public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_TYPE from, final KEY_GENERIC_TYPE
to) { return new UnmodifiableSortedSet KEY_GENERIC_DIAMOND(sortedSet.subSet(from, to)); }

  @Override
  public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) { return new
UnmodifiableSortedSet
 KEY_GENERIC_DIAMOND(sortedSet.headSet(to)); }

  @Override
  public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) { return new
UnmodifiableSortedSet KEY_GENERIC_DIAMOND(sortedSet.tailSet(from)); }

  @Override
  public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return
ITERATORS.unmodifiable(sortedSet.iterator()); }

  @Override
  public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return
ITERATORS.unmodifiable(sortedSet.iterator(from)); }

  @Override
  public KEY_GENERIC_TYPE FIRST() {  return sortedSet.FIRST(); }

  @Override
  public KEY_GENERIC_TYPE LAST() {  return sortedSet.LAST(); }

#if KEYS_PRIMITIVE
  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_CLASS first() {  return sortedSet.first(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public KEY_CLASS last()
```

```
 {  return sortedSet.last(); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_CLASS from, final
KEY_GENERIC_CLASS to) { return new UnmodifiableSortedSet(sortedSet.subSet(from, to)); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_CLASS to) { return new
UnmodifiableSortedSet(sortedSet.headSet(to)); }

  /** {@inheritDoc}
   * @deprecated Please use the corresponding type-specific method instead. */
  @Deprecated
  @Override
  public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_CLASS from) { return new
UnmodifiableSortedSet(sortedSet.tailSet(from)); }
#endif
 }


 /** Returns an unmodifiable type-specific sorted set backed by the given type-specific sorted set.
  *
  * @param s the sorted set to be wrapped in an unmodifiable
 sorted set.
  * @return an unmodifiable view of the specified sorted set.
  * @see java.util.Collections#unmodifiableSortedSet(SortedSet)
  */
 public static KEY_GENERIC SORTED_SET KEY_GENERIC unmodifiable(final SORTED_SET
KEY_GENERIC s) { return new UnmodifiableSortedSet KEY_GENERIC_DIAMOND(s); }



#if defined(TEST) && ! KEY_CLASS_Reference

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif KEY_CLASS_Object
  return Integer.toBinaryString(r.nextInt());
#endif
```

```
 }


 protected static void testSets(KEY_TYPE k, SORTED_SET m, SortedSet t, int level) {
  int n = 100;
  int c;

  long ms;
  boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement, mThrowsIndex, tThrowsIndex,
mThrowsUnsupp, tThrowsUnsupp;
  boolean rt = false, rm = false;

  if (level == 0) return;

  /* Now we check that m and t are equal. */
  if (!m.equals(t) || !
 t.equals(m)) System.err.println("m: " + m + " t: " + t);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");

  /* Now we check that m actually holds that data. */
  for(java.util.Iterator i=t.iterator(); i.hasNext();) {
   ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */
  for(java.util.Iterator i=m.iterator(); i.hasNext();) {
   ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
m)");
  }

  /* Now we check that inquiries about random data give the same answer in m and t. For
     m we use the polymorphic method. */

  for(int i=0; i<n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement
 = mThrowsIllegal = tThrowsIllegal = mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    m.contains(T);
   }
   catch (NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
```

```
  try {
   t.contains(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
  ensure(mThrowsIndex == tThrowsIndex, "Error
 (" + level + ", " + seed + "): contains() divergence in IndexOutOfBoundsException for " + T + " (" + mThrowsIndex
+ ", " + tThrowsIndex + ") " + m);
  if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex) ensure(m.contains(KEY2OBJ(T)) ==
t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in keys between t and m (polymorphic
method) " + m);
 }

 /* Again, we check that inquiries about random data give the same answer in m and t, but
    for m we use the standard method. */

 for(int i=0; i<n;  i++) {
  KEY_TYPE T = genKey();

  mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

  try {
   m.contains(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
  catch (UnsupportedOperationException e) { mThrowsUnsupp
 = true; }

  try {
   t.contains(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
  catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
```

```
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " +
 seed + "): contains() divergence in UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " +
tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp)
ensure(m.contains(KEY2OBJ(T)) ==  t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence
between t and m (standard method) " + m);
  }

  /* Now we add and remove random data in m and t, checking that the result is the same. */

  for(int i=0; i<20*n;  i++) {
   KEY_TYPE T = genKey();

   mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

   try {
    rm = m.add(KEY2OBJ(T));
   }
   catch (NoSuchElementException e) { mThrowsNoElement = true; }
   catch (IllegalArgumentException e) { mThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
   catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

   try {
    rt = t.add(KEY2OBJ(T));
   }
   catch
 (NoSuchElementException e) { tThrowsNoElement = true; }
   catch (IllegalArgumentException e) { tThrowsIllegal = true; }
   catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
   catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

   ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
   ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
   ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): add() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
   ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): add() divergence in
UnsupportedOperationException
 for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
   if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in add() between t and m " + m);
```

```
  T = genKey();

  mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

  try {
   rm = m.remove(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
  catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

  try {
   rt = t.remove(KEY2OBJ(T));
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
  catch (UnsupportedOperationException
 e) { tThrowsUnsupp = true; }

  if (! KEY_EQUALS(T, k) && ! mThrowsUnsupp && tThrowsUnsupp) mThrowsUnsupp = false; // Stupid bug in
Collections.singleton()

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
  ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): remove() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
  // This cannot be possibly made work without a lot of fuss
  //ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): remove() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " +
 tThrowsUnsupp + ") " + m);
  if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in remove() between t and m " + m);
  }

 ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal " + m);
 ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal " + m);

 /* Now we add and remove random collections in m and t, checking that the result is the same. */

 for(int i=0; i<20*n;  i++) {
  KEY_TYPE T = genKey();
```

```
  mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

  try {
   rm = m.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
  catch (UnsupportedOperationException
 e) { mThrowsUnsupp = true; }

  try {
   rt = t.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
  }
  catch (NoSuchElementException e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
  catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): addAll() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): addAll() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
  ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): addAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
  ensure(mThrowsUnsupp
 == tThrowsUnsupp, "Error (" + level + ", " + seed + "): addAll() divergence in UnsupportedOperationException for
" + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
  if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in addAll() between t and m " + m);

  T = genKey();

  mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

  try {
   rm = m.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
  }
  catch (NoSuchElementException e) { mThrowsNoElement = true; }
  catch (IllegalArgumentException e) { mThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
  catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

  try {
   rt = t.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
  }
```

```
  catch (NoSuchElementException
 e) { tThrowsNoElement = true; }
  catch (IllegalArgumentException e) { tThrowsIllegal = true; }
  catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
  catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }


  ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): removeAll() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
  ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): removeAll() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
  ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): removeAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
  ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): removeAll() divergence in
UnsupportedOperationException for " + T + " (" +
 mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
  if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in removeAll() between t and m " + m);
  }

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after set removal " + m);
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after set removal " + m);

  /* Now we check that m actually holds the same data. */

  for(java.util.Iterator i=t.iterator(); i.hasNext();) {
  ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
t)");
  }

  /* Now we check that m actually holds that data, but iterating on m. */

  for(java.util.Iterator i=m.iterator(); i.hasNext();) {
  ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
m)");
  }

  if (m instanceof Singleton) {
  ensure(m.equals(((Singleton)m).clone()),
 "Error (" + level + ", " + seed + "): m does not equal m.clone()");
  ensure(((Singleton)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
  }

  int h = m.hashCode();

  /* Now we save and read m. */

  SORTED_SET m2 = null;
```

```
  try {
   java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test." + m.getClass().getSimpleName() + "." + n);
   java.io.OutputStream os = new java.io.FileOutputStream(ff);
   java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

   oos.writeObject(m);
   oos.close();

   java.io.InputStream is = new java.io.FileInputStream(ff);
   java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

   m2 = (SORTED_SET)ois.readObject();
   ois.close();
   ff.delete();
  }
  catch(Exception e) {
   e.printStackTrace();
   System.exit(1);
  }

#if ! KEY_CLASS_Reference

  ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

  /*
 Now we check that m2 actually holds that data. */

  ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
  ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
#endif

  /* Now we select a pair of keys and create a subset. */

  if (! m.isEmpty()) {
   java.util.ListIterator i;
   Object start = m.first(), end = m.first();
   for(i = (java.util.ListIterator)m.iterator(); i.hasNext() && r.nextBoolean(); start = end = i.next());
   for(; i.hasNext() && r.nextBoolean(); end = i.next());

   //System.err.println("Checking subSet from " + start + " to " + end + " (level=" + (level+1) + ")...");
   testSets(k, (SORTED_SET)m.subSet((KEY_CLASS)start, (KEY_CLASS)end), t.subSet(start, end), level - 1);

   ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after subSet");
   ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subSet");

   //System.err.println("Checking
```

```
headSet to " + end + " (level=" + (level+1) + ")...");
  testSets(k, (SORTED_SET)m.headSet((KEY_CLASS)end), t.headSet(end), level - 1);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after headSet");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after headSet");

  //System.err.println("Checking tailSet from " + start + " (level=" + (level+1) + ")...");
  testSets(k, (SORTED_SET)m.tailSet((KEY_CLASS)start), t.tailSet(start), level - 1);

  ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after tailSet");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after tailSet");
 }

 return;
}

private static void test() {
 KEY_TYPE k = genKey();
 Singleton m = new Singleton(k);
 SortedSet u = new java.util.TreeSet();
 u.add(KEY2OBJ(k));
 testSets(k, m, java.util.Collections.unmodifiableSortedSet(u), 2);
 System.out.println("Test OK");
}


private static long
seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

private static String format(double d) {
 StringBuffer s = new StringBuffer();
 return format.format(d, s, fp).toString();
}

private static void fatal(String msg) {
 System.out.println(msg);
 System.exit(1);
}

private static void ensure(boolean cond, String msg) {
 if (cond) return;
 fatal(msg);
}
```

```
 /** This method expects as first argument a lower-cased type (e.g., "int"),
  * and as second optional argument a seed. */

 public static void main(String arg[]) throws Exception {
  if (arg.length > 1) r = new java.util.Random(seed = Long.parseLong(arg[1]));

  try {
   test();
  } catch(Throwable e) {
   e.printStackTrace(System.err);
   System.err.println("seed: " + seed);
  }
 }

#endif

}
```

Found in path(s):
*
 /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/SortedSets.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.List;

#if ! KEY_CLASS_Reference

/** A type-specific {@link List}; provides some additional methods that use polymorphism to avoid (un)boxing.
```

```
*
* <p>Note that this type-specific interface extends {@link Comparable}: it is expected that implementing
* classes perform a lexicographical
 comparison using the standard operator "less then" for primitive types,
* and the usual {@link Comparable#compareTo(Object) compareTo()} method for objects.
*
* <p>Additionally, this interface strengthens {@link #listIterator()},
* {@link #listIterator(int)} and {@link #subList(int,int)}.
*
* <p>Besides polymorphic methods, this interfaces specifies methods to copy into an array or remove contiguous
* sublists. Although the abstract implementation of this interface provides simple, one-by-one implementations
* of these methods, it is expected that concrete implementation override them with optimized versions.
*
* @see List
*/

public interface LIST KEY_GENERIC extends List<KEY_GENERIC_CLASS>, Comparable<List<? extends
KEY_GENERIC_CLASS>>, COLLECTION KEY_GENERIC {
#else

/** A type-specific {@link List}; provides some additional methods that use polymorphism to avoid (un)boxing.
*
* <p>Additionally, this interface strengthens {@link #iterator()}, {@link #listIterator()},
 * {@link #listIterator(int)} and {@link #subList(int,int)}. The former had been already
* strengthened upstream, but unfortunately {@link List} re-specifies it.
*
* <p>Besides polymorphic methods, this interfaces specifies methods to copy into an array or remove contiguous
* sublists. Although the abstract implementation of this interface provides simple, one-by-one implementations
* of these methods, it is expected that concrete implementation override them with optimized versions.
*
* @see List
*/

public interface LIST KEY_GENERIC extends List<KEY_GENERIC_CLASS>, COLLECTION KEY_GENERIC
{
#endif

 /** Returns a type-specific iterator on the elements of this list.
  *
  * <p>Note that this specification strengthens the one given in {@link List#iterator()}.
  * It would not be normally necessary, but {@link java.lang.Iterable#iterator()} is bizarrily re-specified
  * in {@link List}.
  *
  * @return an iterator on the elements of this list.
  */
 @Override
 KEY_LIST_ITERATOR
```

```
KEY_GENERIC iterator();

/** Returns a type-specific list iterator on the list.
 *
 * @see List#listIterator()
 */
@Override
KEY_LIST_ITERATOR KEY_GENERIC listIterator();

/** Returns a type-specific list iterator on the list starting at a given index.
 *
 * @see List#listIterator(int)
 */
@Override
KEY_LIST_ITERATOR KEY_GENERIC listIterator(int index);

/** Returns a type-specific view of the portion of this list from the index {@code from}, inclusive, to the index
{@code to}, exclusive.
 *
 * <p>Note that this specification strengthens the one given in {@link List#subList(int,int)}.
 *
 * @see List#subList(int,int)
 */
@Override
LIST KEY_GENERIC subList(int from, int to);

/** Sets the size of this list.
 *
 * <p>If the specified size is smaller than the current size, the last elements are
 * discarded. Otherwise, they are filled with 0/{@code null}/{@code false}.
 *
 * @param size the new size.
 */

void size(int size);

/** Copies (hopefully quickly)
elements of this type-specific list into the given array.
 *
 * @param from the start index (inclusive).
 * @param a the destination array.
 * @param offset the offset into the destination array where to store the first element copied.
 * @param length the number of elements to be copied.
 */
void getElements(int from, KEY_TYPE a[], int offset, int length);

/** Removes (hopefully quickly) elements of this type-specific list.
 *
```

```
 * @param from the start index (inclusive).
 * @param to the end index (exclusive).
 */
void removeElements(int from, int to);

/** Add (hopefully quickly) elements to this type-specific list.
 *
 * @param index the index at which to add elements.
 * @param a the array containing the elements.
 */
void addElements(int index, KEY_GENERIC_TYPE a[]);

/** Add (hopefully quickly) elements to this type-specific list.
 *
 * @param index the index at which to add elements.
 * @param a the array containing the elements.
 * @param offset the offset
of the first element to add.
 * @param length the number of elements to add.
 */
void addElements(int index, KEY_GENERIC_TYPE a[], int offset, int length);

/** Set (hopefully quickly) elements to match the array given.
 * @param a the array containing the elements.
 * @since 8.3.0
 */
default void setElements(KEY_GENERIC_TYPE a[]) {
 setElements(0, a);
}

/** Set (hopefully quickly) elements to match the array given.
 * @param index the index at which to start setting elements.
 * @param a the array containing the elements.
 * @since 8.3.0
 */
default void setElements(int index, KEY_GENERIC_TYPE a[]) {
 setElements(index, a, 0, a.length);
}

/** Set (hopefully quickly) elements to match the array given.
 *
 * Sets each in this list to the corresponding elements in the array, as if by
 * <pre><code>
 * ListIterator iter = listIterator(index);
 * int i = 0;
 * while (i < length) {
 *   iter.next();
 *   iter.set(a[offset + i++]);
```

```
 * }
 * </code></pre>

* However, the exact implementation may be more efficient, taking into account
 * whether random access is faster or not, or at the discretion of subclasses,
 * abuse internals.
 *
 * @param index the index at which to start setting elements.
 * @param a the array containing the elements
 * @param offset the offset of the first element to add.
 * @param length the number of elements to add.
 * @since 8.3.0
 */
default void setElements(int index, KEY_GENERIC_TYPE a[], int offset, int length) {
 // We can't use AbstractList#ensureIndex, sadly.
 if (index < 0)  throw new IndexOutOfBoundsException("Index (" + index + ") is negative");
 if (index > size()) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than list size (" +
(size()) + ")");
 ARRAYS.ensureOffsetLength(a, offset, length);
 if (index + length > size()) throw new IndexOutOfBoundsException("End index (" + (index + length) + ") is greater
than list size (" + size() + ")");
 KEY_LIST_ITERATOR KEY_GENERIC
iter = listIterator(index);
 int i = 0;
 while (i < length) {
  iter.NEXT_KEY();
  iter.set(a[offset + i++]);
 }
}

#if KEYS_PRIMITIVE

/** Appends the specified element to the end of this list (optional operation).
 * @see List#add(Object)
 */
@Override
boolean add(KEY_TYPE key);

/** Inserts the specified element at the specified position in this list (optional operation).
 * @see List#add(int,Object)
 */
void add(int index, KEY_TYPE key);

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default void add(int index, KEY_CLASS key) {
```

```
 add(index, KEY_CLASS2TYPE(key));
}

/** Inserts all of the elements in the specified type-specific collection into this type-specific list at the specified position (optional operation).
 * @see List#addAll(int,java.util.Collection)
 */
boolean addAll(int index, COLLECTION c);

/** Inserts all of the elements in the specified type-specific list into
this type-specific list at the specified position (optional operation).
 * @see List#add(int,Object)
 */
boolean addAll(int index, LIST c);

/** Appends all of the elements in the specified type-specific list to the end of this type-specific list (optional operation).
 * @see List#add(int,Object)
 */
boolean addAll(LIST c);

/** Replaces the element at the specified position in this list with the specified element (optional operation).
 * @see List#set(int,Object)
 */
KEY_TYPE set(int index, KEY_TYPE k);

/** Returns the element at the specified position in this list.
 * @see List#get(int)
 */
KEY_TYPE GET_KEY(int index);

/** Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.
 * @see List#indexOf(Object)
 */
int indexOf(KEY_TYPE k);

/** Returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element.
 * @see List#lastIndexOf(Object)

 */
int lastIndexOf(KEY_TYPE k);

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
```

```
@Override
default boolean contains(final Object key) {
 return COLLECTION.super.contains(key);
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS get(int index) {
 return KEY2OBJ(GET_KEY(index));
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default int indexOf(Object o) {
 return indexOf(KEY_OBJ2TYPE(o));
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default int lastIndexOf(Object o) {
 return lastIndexOf(KEY_OBJ2TYPE(o));
}

/** {@inheritDoc}
 * <p>This method specification is a workaround for
 * <a href="http://bugs.java.com/bugdatabase/view_bug.do?bug_id=JDK-8177440">bug
8177440</a>.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default boolean add(KEY_CLASS k) {
 return add(KEY_CLASS2TYPE(k));
}

/** Removes the element at the specified position in this list (optional operation).
 * @see List#remove(int)
 */
KEY_TYPE REMOVE_KEY(int index);

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
```

```
@Deprecated
@Override
default boolean remove(final Object key) {
 return COLLECTION.super.remove(key);
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS remove(int index) {
 return KEY2OBJ(REMOVE_KEY(index));
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS set(int index, KEY_CLASS k) {
 return KEY2OBJ(set(index, KEY_CLASS2TYPE(k)));
}

#endif

#if
 defined KEY_COMPARATOR && KEYS_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default void sort(final java.util.Comparator<? super KEY_GENERIC_CLASS> comparator) {
 sort(COMPARATORS.AS_KEY_COMPARATOR(comparator));
}

/** Sort a list using a type-specific comparator.
 *
 * <p>Pass {@code null} to sort using natural ordering.
 * @see List#sort(java.util.Comparator)
 *
 * @implSpec The default implementation dumps the elements into an array using
 * {@link #toArray()}, sorts the array, then replaces all elements using the
 * {@link #setElements} function.
 *
 * @implNote It is possible for this method to call {@link #unstableSort} if it can
 * determine that the results of a stable and unstable sort are completely equivalent.
 * This means if you override {@link #unstableSort}, it should <em>not</em> call this
 * method unless you override this method as well.
```

```
 *
 * @since
8.3.0
 */
 default void sort(final KEY_COMPARATOR comparator) {
#if !(KEY_CLASS_Float || KEY_CLASS_Double)
  if (comparator == null) {
   // For non-floating point primitive types, when comparing naturally,
   // it is impossible to tell the difference between a stable and not-stable sort.
   // So just use the probably faster unstable sort.
   unstableSort(comparator);
  } else {
   KEY_TYPE[] elements = TO_KEY_ARRAY();
   ARRAYS.stableSort(elements, comparator);
   setElements(elements);
  }
#else
  KEY_TYPE[] elements = TO_KEY_ARRAY();
  if (comparator == null) {
   ARRAYS.stableSort(elements);
  } else {
   ARRAYS.stableSort(elements, comparator);
  }
  setElements(elements);
#endif
 }

 /** Sorts this list using a sort not assured to be stable.
  * @deprecated Please use the corresponding type-specific method instead.
  */
 @Deprecated
 default void unstableSort(final java.util.Comparator<? super KEY_GENERIC_CLASS> comparator) {
  unstableSort(COMPARATORS.AS_KEY_COMPARATOR(comparator));
 }

 /**
 Sorts this list using a sort not assured to be stable.
  *
  * <p>Pass {@code null} to sort using natural ordering.
  *
  * <p>This differs from {@link List#sort(java.util.Comparator)} in that the results are
  * not assured to be stable, but may be a bit faster.
  *
  * <p>Unless a subclass specifies otherwise, the results of the method if the list is
  * concurrently modified during the sort are unspecified.
  *
  * @implSpec The default implementation dumps the elements into an array using
  * {@link #toArray()}, sorts the array, then replaces all elements using the
```

```
  * {@link #setElements} function.
  *
  * @since 8.3.0
  */
 default void unstableSort(final KEY_COMPARATOR comparator) {
  KEY_TYPE[] elements = TO_KEY_ARRAY();
  if (comparator == null) {
   ARRAYS.unstableSort(elements);
  } else {
   ARRAYS.unstableSort(elements, comparator);
  }
  setElements(elements);
 }
#else
#if !KEYS_REFERENCE
#error Assertion error: No KEY_COMPARATOR defined, but not a reference
 type.
#endif
 /** Sorts this list using a sort not assured to be stable.
  * This differs from {@link List#sort(java.util.Comparator)} in that the results are
  * not assured to be stable, but may be a bit faster.
  *
  * <p>Unless a subclass specifies otherwise, the results of the method if the list is
  * concurrently modified during the sort are unspecified.
  *
  * @implSpec The default implementation dumps the elements into an array using
  * {@link #toArray()}, sorts the array, then replaces all elements using the
  * {@link #setElements} function.
  *
  * @since 8.3.0
  */
 SUPPRESS_WARNINGS_KEY_UNCHECKED
 default void unstableSort(final java.util.Comparator<? super KEY_GENERIC_CLASS> comparator) {
  KEY_GENERIC_TYPE[] elements = (KEY_GENERIC_TYPE[])toArray();
  if (comparator == null) {
   ARRAYS.unstableSort(elements);
  } else {
   ARRAYS.unstableSort(elements, comparator);
  }
  setElements(elements);
 }
#endif
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/List.drv

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

/** An abstract class providing basic methods for sorted sets implementing a type-specific interface. */

public abstract class ABSTRACT_SORTED_SET KEY_GENERIC extends ABSTRACT_SET KEY_GENERIC
implements SORTED_SET KEY_GENERIC {

 protected ABSTRACT_SORTED_SET() {}

 @Override
 public abstract KEY_BIDI_ITERATOR
 KEY_GENERIC iterator();
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractSortedSet.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import it.unimi.dsi.fastutil.BidirectionalIterator;
#if KEYS_PRIMITIVE
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;
#endif

/** A type-specific bidirectional iterator; provides an additional method to avoid (un)boxing,
* and the possibility to skip elements backwards.
*
* @see
 BidirectionalIterator
*/

#if KEYS_PRIMITIVE
public interface KEY_BIDI_ITERATOR KEY_GENERIC extends KEY_ITERATOR KEY_GENERIC,
ObjectBidirectionalIterator<KEY_GENERIC_CLASS> {
#else
public interface KEY_BIDI_ITERATOR KEY_GENERIC extends KEY_ITERATOR KEY_GENERIC,
BidirectionalIterator<KEY_GENERIC_CLASS> {
#endif

#if KEYS_PRIMITIVE

 /**
  * Returns the previous element as a primitive type.
  *
  * @return the previous element in the iteration.
  * @see java.util.ListIterator#previous()
  */

 KEY_TYPE PREV_KEY();

 /** {@inheritDoc}
  * @deprecated Please use the corresponding type-specific method instead. */
 @Deprecated
 @Override
 default KEY_GENERIC_CLASS previous() { return KEY_CLASS.valueOf(PREV_KEY()); }

#endif
```

```
 /** Moves back for the given number of elements.
  *
  * <p>The effect of this call is exactly the same as that of
  * calling {@link #previous()} for {@code n} times (possibly stopping
  * if {@link #hasPrevious()} becomes false).
  *
  * @param n the number of elements
 to skip back.
  * @return the number of elements actually skipped.
  * @see #previous()
  */
#if KEYS_PRIMITIVE
 @Override
#endif
 default int back(final int n) {
  int i = n;
  while(i-- != 0 && hasPrevious()) PREV_KEY();
  return n - i - 1;
 }

 /** {@inheritDoc} */
 @Override
 default int skip(final int n) {
  return KEY_ITERATOR.super.skip(n);
 }
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/BidirectionalIterator.drv
No license file was found, but licenses were detected in source scan.


```
/*
* Copyright (C) 2002-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package PACKAGE;

/**  An abstract class facilitating the creation of type-specific {@linkplain java.util.Comparator comparators}.
*
* @see java.util.Comparator
* @deprecated As of fastutil 8 this class is no longer necessary, as its only previous abstract
* method is now a default method of the type-specific interface.
 */

@Deprecated
public abstract class KEY_ABSTRACT_COMPARATOR KEY_GENERIC implements KEY_COMPARATOR
KEY_GENERIC, java.io.Serializable {
 private static final long serialVersionUID = 0L;
 protected KEY_ABSTRACT_COMPARATOR() {}
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/AbstractComparator.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007-2020 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

import java.util.Map;
import java.util.NoSuchElementException;
import it.unimi.dsi.fastutil.objects.AbstractObjectSet;
import it.unimi.dsi.fastutil.objects.ObjectIterator;

import VALUE_PACKAGE.VALUE_COLLECTION;
import VALUE_PACKAGE.VALUE_ABSTRACT_COLLECTION;
```

```
import VALUE_PACKAGE.VALUE_ARRAYS;

/** A
 simple, brute-force implementation of a map based on two parallel backing arrays.
*
* <p>The main purpose of this
* implementation is that of wrapping cleanly the brute-force approach to the storage of a very
* small number of pairs: just put them into two parallel arrays and scan linearly to find an item.
*/

public class ARRAY_MAP KEY_VALUE_GENERIC extends ABSTRACT_MAP KEY_VALUE_GENERIC
implements java.io.Serializable, Cloneable {

 private static final long serialVersionUID = 1L;
 /** The keys (valid up to {@link #size}, excluded). */
 private transient KEY_TYPE[] key;
 /** The values (parallel to {@link #key}). */
 private transient VALUE_TYPE[] value;
 /** The number of valid entries in {@link #key} and {@link #value}. */
 private int size;

 /** Creates a new empty array map with given key and value backing arrays. The resulting map will have as many
entries as the given arrays.
  *
  * <p>It is responsibility of the caller that the elements of {@code key} are distinct.
  *

 * @param key the key array.
  * @param value the value array (it <em>must</em> have the same length as {@code key}).
  */
 public ARRAY_MAP(final KEY_TYPE[] key, final VALUE_TYPE[] value) {
  this.key = key;
  this.value = value;
  size = key.length;
  if(key.length != value.length) throw new IllegalArgumentException("Keys and values have different lengths (" +
key.length + ", " + value.length + ")");
 }

 /** Creates a new empty array map.
  */
 public ARRAY_MAP() {
  this.key = ARRAYS.EMPTY_ARRAY;
  this.value = VALUE_ARRAYS.EMPTY_ARRAY;
 }

 /** Creates a new empty array map of given capacity.
  *
  * @param capacity the initial capacity.
```

```
 */
public ARRAY_MAP(final int capacity) {
 this.key = new KEY_TYPE[capacity];
 this.value = new VALUE_TYPE[capacity];
}

/** Creates a new empty array map copying the entries of a given map.
 *
 * @param m a map.
 */
public ARRAY_MAP(final MAP KEY_VALUE_GENERIC m) {
 this(m.size());
 int i = 0;
 for(MAP.Entry KEY_VALUE_GENERIC e :
m.ENTRYSET()) {
  key[i] = e.ENTRY_GET_KEY();
  value[i] = e.ENTRY_GET_VALUE();
  i++;
 }
 size = i;
}

/** Creates a new empty array map copying the entries of a given map.
 *
 * @param m a map.
 */
public ARRAY_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> m)
{
 this(m.size());
 int i = 0;
 for(Map.Entry<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> e : m.entrySet()) {
  key[i] = KEY_CLASS2TYPE(e.getKey());
  value[i] = VALUE_CLASS2TYPE(e.getValue());
  i++;
 }
 size = i;
}

/** Creates a new array map with given key and value backing arrays, using the given number of elements.
 *
 * <p>It is responsibility of the caller that the first {@code size} elements of {@code key} are distinct.
 *
 * @param key the key array.
 * @param value the value array (it <em>must</em> have the same length as {@code key}).
 * @param size the number of valid elements in {@code key} and {@code value}.
 */
public ARRAY_MAP(final KEY_TYPE[]
```

```
 key, final VALUE_TYPE[] value, final int size) {
  this.key = key;
  this.value = value;
  this.size = size;
  if(key.length != value.length) throw new IllegalArgumentException("Keys and values have different lengths (" +
key.length + ", " + value.length + ")");
  if (size > key.length) throw new IllegalArgumentException("The provided size (" + size + ") is larger than or equal
to the backing-arrays size (" + key.length + ")");
 }

 private final class EntrySet extends AbstractObjectSet<MAP.Entry KEY_VALUE_GENERIC> implements
FastEntrySet KEY_VALUE_GENERIC {

  @Override
  public ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() {
   return new ObjectIterator<MAP.Entry KEY_VALUE_GENERIC>() {
    int curr = -1, next = 0;

    @Override
    public boolean hasNext() { return next < size; }

    @Override
    SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
    public Entry KEY_VALUE_GENERIC next() {
     if (! hasNext()) throw new NoSuchElementException();
     return new ABSTRACT_MAP.BasicEntry
 KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST key[curr = next], VALUE_GENERIC_CAST
value[next++]);
    }

    @Override
    public void remove() {
     if (curr == -1) throw new IllegalStateException();
     curr = -1;
     final int tail = size-- - next--;
     System.arraycopy(key, next + 1, key, next, tail);
     System.arraycopy(value, next + 1, value, next, tail);
#if KEYS_REFERENCE
     key[size] = null;
#endif
#if VALUES_REFERENCE
     value[size] = null;
#endif
    }
   };
  }

  @Override
```

```
  public ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> fastIterator() {
   return new ObjectIterator<MAP.Entry KEY_VALUE_GENERIC>() {
    int next = 0, curr = -1;
    final BasicEntry KEY_VALUE_GENERIC entry = new BasicEntry KEY_VALUE_GENERIC_DIAMOND ();

    @Override
    public boolean hasNext() { return next < size; }

    @Override
    SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
    public Entry KEY_VALUE_GENERIC next() {
     if (! hasNext()) throw new NoSuchElementException();
     entry.key
 = KEY_GENERIC_CAST key[curr = next];
     entry.value = VALUE_GENERIC_CAST value[next++];
     return entry;
    }

    @Override
    public void remove() {
     if (curr == -1) throw new IllegalStateException();
     curr = -1;
     final int tail = size-- - next--;
     System.arraycopy(key, next + 1, key, next, tail);
     System.arraycopy(value, next + 1, value, next, tail);
#if KEYS_REFERENCE
     key[size] = null;
#endif
#if VALUES_REFERENCE
     value[size] = null;
#endif
    }
   };
  }

  @Override
  public int size() { return size; }

  @Override
  SUPPRESS_WARNINGS_KEY_UNCHECKED
  public boolean contains(Object o) {
   if (! (o instanceof Map.Entry)) return false;
   final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
   if (e.getKey() == null || ! (e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
   if (e.getValue() == null || ! (e.getValue() instanceof VALUE_CLASS)) return false;
```

```
#endif
   final KEY_GENERIC_TYPE k = KEY_OBJ2TYPE(KEY_GENERIC_CAST
 e.getKey());
   return ARRAY_MAP.this.containsKey(k) && VALUE_EQUALS(ARRAY_MAP.this.GET_VALUE(k),
VALUE_OBJ2TYPE(e.getValue()));
  }

  @Override
  SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
  public boolean remove(final Object o) {
   if (!(o instanceof Map.Entry)) return false;
   final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
   if (e.getKey() == null || ! (e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
   if (e.getValue() == null || ! (e.getValue() instanceof VALUE_CLASS)) return false;
#endif
   final KEY_GENERIC_TYPE k = KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey());
   final VALUE_GENERIC_TYPE v = VALUE_OBJ2TYPE(VALUE_GENERIC_CAST e.getValue());

   final int oldPos = ARRAY_MAP.this.findKey(k);
   if (oldPos == -1 || ! VALUE_EQUALS(v, ARRAY_MAP.this.value[oldPos])) return false;
   final int tail = size - oldPos - 1;
   System.arraycopy(ARRAY_MAP.this.key, oldPos + 1, ARRAY_MAP.this.key, oldPos, tail);
   System.arraycopy(ARRAY_MAP.this.value,
 oldPos + 1, ARRAY_MAP.this.value, oldPos, tail);
   ARRAY_MAP.this.size--;
#if KEYS_REFERENCE
   ARRAY_MAP.this.key[size] = null;
#endif
#if VALUES_REFERENCE
   ARRAY_MAP.this.value[size] = null;
#endif
   return true;
  }
 }

 @Override
 public FastEntrySet KEY_VALUE_GENERIC ENTRYSET() { return new EntrySet(); }

 private int findKey(final KEY_TYPE k) {
  final KEY_TYPE[] key = this.key;
  for(int i = size; i-- != 0;) if (KEY_EQUALS(key[i], k)) return i;
  return -1;
 }

 @Override
```

```
SUPPRESS_WARNINGS_VALUE_UNCHECKED
public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
 final KEY_TYPE[] key = this.key;
 for(int i = size; i-- != 0;) if (KEY_EQUALS(key[i], k)) return VALUE_GENERIC_CAST value[i];
 return defRetValue;
}

@Override
public int size() { return size; }

@Override
public void clear() {
#if KEYS_REFERENCE || VALUES_REFERENCE
  for(int i = size; i-- != 0;) {
#if KEYS_REFERENCE
   key[i] = null;
#endif
#if VALUES_REFERENCE
   value[i] = null;
#endif
  }
#endif
  size
 = 0;
 }

@Override
public boolean containsKey(final KEY_TYPE k) { return findKey(k) != -1; }

@Override
public boolean containsValue(VALUE_TYPE v) {
 for(int i = size; i-- != 0;) if (VALUE_EQUALS(value[i], v)) return true;
 return false;
}

@Override
public boolean isEmpty() { return size == 0; }

@Override
SUPPRESS_WARNINGS_VALUE_UNCHECKED
public VALUE_GENERIC_TYPE put(KEY_GENERIC_TYPE k, VALUE_GENERIC_TYPE v) {
 final int oldKey = findKey(k);
 if (oldKey != -1) {
  final VALUE_GENERIC_TYPE oldValue = VALUE_GENERIC_CAST value[oldKey];
  value[oldKey] = v;
  return oldValue;
 }
 if (size == key.length) {
```

```
  final KEY_TYPE[] newKey = new KEY_TYPE[size == 0 ? 2 : size * 2];
  final VALUE_TYPE[] newValue = new VALUE_TYPE[size == 0 ? 2 : size * 2];
  for(int i = size; i-- != 0;) {
   newKey[i] = key[i];
   newValue[i] = value[i];
  }
  key = newKey;
  value = newValue;
 }
 key[size] = k;
 value[size] = v;
 size++;
 return defRetValue;
}

@Override
SUPPRESS_WARNINGS_VALUE_UNCHECKED
public
VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) {
 final int oldPos = findKey(k);
 if (oldPos == -1) return defRetValue;
 final VALUE_GENERIC_TYPE oldValue = VALUE_GENERIC_CAST value[oldPos];
 final int tail = size - oldPos - 1;
 System.arraycopy(key, oldPos + 1, key, oldPos, tail);
 System.arraycopy(value, oldPos + 1, value, oldPos, tail);
 size--;
#if KEYS_REFERENCE
 key[size] = null;
#endif
#if VALUES_REFERENCE
 value[size] = null;
#endif
 return oldValue;
}

@Override
public SET KEY_GENERIC keySet() {
 return new ABSTRACT_SET KEY_GENERIC() {
  @Override
  public boolean contains(final KEY_TYPE k) {
   return findKey(k) != -1;
  }

  @Override
  public boolean remove(final KEY_TYPE k) {
    final int oldPos = findKey(k);
    if (oldPos == -1) return false;
    final int tail = size - oldPos - 1;
```

```
    System.arraycopy(key, oldPos + 1, key, oldPos, tail);
    System.arraycopy(value, oldPos + 1, value, oldPos, tail);

size--;
    return true;
  }

  @Override
  public KEY_ITERATOR KEY_GENERIC iterator() {
   return new KEY_ITERATOR KEY_GENERIC() {
    int pos = 0;
    @Override
    public boolean hasNext() {
     return pos < size;
    }

    @Override
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public KEY_GENERIC_TYPE NEXT_KEY() {
     if (! hasNext()) throw new NoSuchElementException();
     return KEY_GENERIC_CAST key[pos++];
    }

    @Override
    public void remove() {
     if (pos == 0) throw new IllegalStateException();
     final int tail = size - pos;
     System.arraycopy(key, pos, key, pos - 1, tail);
     System.arraycopy(value, pos, value, pos - 1, tail);
     size--;
     pos--;
    }
   };
  }

  @Override
  public int size() {
   return size;
  }

  @Override
  public void clear() {
   ARRAY_MAP.this.clear();
  }
 };
}

@Override
```

```
public VALUE_COLLECTION VALUE_GENERIC values() {
 return new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {

  @Override
  public
boolean contains(final VALUE_TYPE v) {
   return containsValue(v);
  }

  @Override
  public VALUE_PACKAGE.VALUE_ITERATOR VALUE_GENERIC iterator() {
   return new VALUE_PACKAGE.VALUE_ITERATOR VALUE_GENERIC() {
    int pos = 0;
    @Override
    public boolean hasNext() {
     return pos < size;
    }

    @Override
    SUPPRESS_WARNINGS_VALUE_UNCHECKED
    public VALUE_GENERIC_TYPE NEXT_VALUE() {
     if (! hasNext()) throw new NoSuchElementException();
     return VALUE_GENERIC_CAST value[pos++];
    }

    @Override
    public void remove() {
     if (pos == 0) throw new IllegalStateException();
     final int tail = size - pos;
     System.arraycopy(key, pos, key, pos - 1, tail);
     System.arraycopy(value, pos, value, pos - 1, tail);
     size--;
    }
   };
  }

  @Override
  public int size() {
   return size;
  }

  @Override
  public void clear() {
   ARRAY_MAP.this.clear();
  }
 };
}
```

```
/** Returns a deep copy of this map.

 *
 * <p>This method performs a deep copy of this hash map; the data stored in the
 * map, however, is not cloned. Note that this makes a difference only for object keys.
 *
 *  @return a deep copy of this map.
 */
@Override
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public ARRAY_MAP KEY_VALUE_GENERIC clone() {
 ARRAY_MAP KEY_VALUE_GENERIC c;
 try {
  c = (ARRAY_MAP KEY_VALUE_GENERIC)super.clone();
 }
 catch(CloneNotSupportedException cantHappen) {
  throw new InternalError();
 }
 c.key = key.clone();
 c.value = value.clone();
 return c;
}

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
 s.defaultWriteObject();
 for(int i = 0; i < size; i++) {
  s.WRITE_KEY(key[i]);
  s.WRITE_VALUE(value[i]);
 }
}

private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
 s.defaultReadObject();
 key = new KEY_TYPE[size];
 value = new VALUE_TYPE[size];
 for(int i = 0; i < size; i++) {
  key[i]
= s.READ_KEY();
  value[i] = s.READ_VALUE();
 }
}
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/ArrayMap.drv
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2020 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/


package PACKAGE;

#if KEY_CLASS_Object
import java.util.Comparator;
import it.unimi.dsi.fastutil.IndirectPriorityQueue;
#endif

import it.unimi.dsi.fastutil.ints.IntArrays;

import java.util.NoSuchElementException;

/** A type-specific array-based semi-indirect priority queue.
*
* <p>Instances of this
 class use as reference list a <em>reference array</em>,
* which must be provided to each constructor, and represent a priority queue
* using a backing array of integer indices&mdash;all operations are performed
* directly on the array. The array is enlarged as needed, but it is never
* shrunk. Use the {@link #trim()} method to reduce its size, if necessary.
*
* <p>This implementation is extremely inefficient, but it is difficult to beat
* when the size of the queue is very small. Moreover, it allows to enqueue several
* time the same index, without limitations.
*/

public class ARRAY_INDIRECT_PRIORITY_QUEUE KEY_GENERIC implements
INDIRECT_PRIORITY_QUEUE KEY_GENERIC {

 /** The reference array. */
 protected KEY_GENERIC_TYPE refArray[];
```

```
/** The backing array. */
protected int array[] = IntArrays.EMPTY_ARRAY;

/** The number of elements in this queue. */
protected int size;

/** The type-specific comparator used in this queue. */
protected KEY_COMPARATOR KEY_SUPER_GENERIC
c;

/** The first index, cached, if {@link #firstIndexValid} is true. */
protected int firstIndex;

/** Whether {@link #firstIndex} contains a valid value. */
protected boolean firstIndexValid;


/** Creates a new empty queue without elements with a given capacity and comparator.
 *
 * @param refArray the reference array.
 * @param capacity the initial capacity of this queue.
 * @param c the comparator used in this queue, or {@code null} for the natural order.
 */
public ARRAY_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, int capacity,
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
 if (capacity > 0) this.array = new int[capacity];
 this.refArray = refArray;
 this.c = c;
}

/** Creates a new empty queue with given capacity and using the natural order.
 *
 * @param refArray the reference array.
 * @param capacity the initial capacity of this queue.
 */
public ARRAY_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, int capacity) {
 this(refArray, capacity, null);
}

/**
Creates a new empty queue with capacity equal to the length of the reference array and a given comparator.
 *
 * @param refArray the reference array.
 * @param c the comparator used in this queue, or {@code null} for the natural order.
 */
public ARRAY_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, KEY_COMPARATOR
KEY_SUPER_GENERIC c) {
 this(refArray, refArray.length, c);
```

```
}

/** Creates a new empty queue with capacity equal to the length of the reference array and using the natural order.
 * @param refArray the reference array.
 */
public ARRAY_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray) {
 this(refArray, refArray.length, null);
}


/** Wraps a given array in a queue using a given comparator.
 *
 * <p>The queue returned by this method will be backed by the given array.
 *
 * @param refArray the reference array.
 * @param a an array of indices into {@code refArray}.
 * @param size the number of elements to be included in the queue.
 * @param
c the comparator used in this queue, or {@code null} for the natural order.
 */
public ARRAY_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, int size,
final KEY_COMPARATOR KEY_SUPER_GENERIC c) {
 this(refArray, 0, c);
 this.array = a;
 this.size = size;
}


/** Wraps a given array in a queue using a given comparator.
 *
 * <p>The queue returned by this method will be backed by the given array.
 *
 * @param refArray the reference array.
 * @param a an array of indices into {@code refArray}.
 * @param c the comparator used in this queue, or {@code null} for the natural order.
 */
public ARRAY_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, final
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
 this(refArray, a, a.length, c);
}

/** Wraps a given array in a queue using the natural order.
 *
 * <p>The queue returned by this method will be backed by the given array.
 *
 * @param refArray the reference array.
 * @param a an array
```

```
of indices into {@code refArray}.
 * @param size the number of elements to be included in the queue.
 */
public ARRAY_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, int size) {
 this(refArray, a, size, null);
}


/** Wraps a given array in a queue using the natural order.
 *
 * <p>The queue returned by this method will be backed by the given array.
 *
 * @param refArray the reference array.
 * @param a an array of indices into {@code refArray}.
 */
public ARRAY_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a) {
 this(refArray, a, a.length);
}


/** Returns the index (in {@link #array}) of the smallest element. */

SUPPRESS_WARNINGS_KEY_UNCHECKED
private int findFirst() {
 if (firstIndexValid) return this.firstIndex;
 firstIndexValid = true;
 int i = size;
 int firstIndex = --i;
 KEY_GENERIC_TYPE first = refArray[array[firstIndex]];

 if (c == null) while(i-- != 0) { if (KEY_LESS(refArray[array[i]], first))
first = refArray[array[firstIndex = i]]; }
 else while(i-- != 0) { if (c.compare(refArray[array[i]], first) < 0) first = refArray[array[firstIndex = i]]; }

 return this.firstIndex = firstIndex;
}

/** Returns the index (in {@link #array}) of the largest element. */

SUPPRESS_WARNINGS_KEY_UNCHECKED
private int findLast() {
 int i = size;
 int lastIndex = --i;
 KEY_GENERIC_TYPE last = refArray[array[lastIndex]];

 if (c == null) { while(i-- != 0) if (KEY_LESS(last, refArray[array[i]])) last = refArray[array[lastIndex = i]]; }
 else { while(i-- != 0) if (c.compare(last, refArray[array[i]]) < 0) last = refArray[array[lastIndex = i]]; }
```

```
 return lastIndex;
}

protected final void ensureNonEmpty() {
 if (size == 0) throw new NoSuchElementException();
}

/** Ensures that the given index is a firstIndexValid reference.
 *
 * @param index an index in the reference array.
 * @throws IndexOutOfBoundsException if the given index is negative or larger than the reference array
length.
 */
protected void ensureElement(final int index) {
 if (index < 0)  throw new IndexOutOfBoundsException("Index (" + index + ") is negative");
 if (index >= refArray.length) throw new IndexOutOfBoundsException("Index (" + index + ") is larger than or equal
to reference array size (" + refArray.length + ")");
}

/** {@inheritDoc}
 *
 * <p>Note that for efficiency reasons this method will <em>not</em> throw an exception
 * when {@code x} is already in the queue. However, the queue state will become
 * inconsistent and the following behaviour will not be predictable.
 */
@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public void enqueue(int x) {
 ensureElement(x);

 if (size == array.length) array = IntArrays.grow(array, size + 1);
 if (firstIndexValid) {
  if (c == null) { if (KEY_LESS(refArray[x], refArray[array[firstIndex]])) firstIndex = size; }
  else if (c.compare(refArray[x], refArray[array[firstIndex]]) < 0) firstIndex = size;
 }
 else firstIndexValid = false;
 array[size++]
= x;
}

@Override
public int dequeue() {
 ensureNonEmpty();
 final int firstIndex = findFirst();
 final int result = array[firstIndex];
 if (--size != 0) System.arraycopy(array, firstIndex + 1, array, firstIndex, size - firstIndex);
 firstIndexValid = false;
 return result;
```

```
}

@Override
public int first() {
 ensureNonEmpty();
 return array[findFirst()];
}

@Override
public int last() {
 ensureNonEmpty();
 return array[findLast()];
}

@Override
public void changed() {
 ensureNonEmpty();
 firstIndexValid = false;
}

/** {@inheritDoc}
 *
 * <p>Note that for efficiency reasons this method will <em>not</em> throw an exception
 * when {@code index} is not in the queue.
 */
@Override
public void changed(int index) {
 ensureElement(index);
 if (index == firstIndex) firstIndexValid = false;
}

/** Signals the queue that all elements have changed. */
@Override
public void allChanged() { firstIndexValid = false; }

@Override
public boolean
remove(int index) {
 ensureElement(index);
 final int[] a = array;
 int i = size;
 while(i-- != 0) if (a[i] == index) break;
 if (i < 0) return false;
 firstIndexValid = false;
 if (--size != 0) System.arraycopy(a, i + 1, a, i, size - i);
 return true;
}
```

```
/** Writes in the provided array the <em>front</em> of the queue, that is, the set of indices
 * whose elements have the same priority as the top.
 * @param a an array whose initial part will be filled with the frnot (must be sized as least as the heap size).
 * @return the number of elements of the front.
 */
@Override
public int front(int[] a) {
 final KEY_GENERIC_TYPE top = refArray[array[findFirst()]];
 int i = size, c = 0;
 while(i-- != 0) if (KEY_EQUALS_NOT_NULL(top, refArray[array[i]])) a[c++] = array[i];
 return c;
}

@Override
public int size() { return size; }

@Override
public void clear() { size = 0; firstIndexValid = false; }

/** Trims the backing array so that it has exactly {@link #size()}
elements. */
public void trim() {
 array = IntArrays.trim(array, size);
}

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return c; }

@Override
public String toString() {
 StringBuffer s = new StringBuffer();
 s.append("[");
 for (int i = 0; i < size; i++) {
  if (i != 0) s.append(", ");
  s.append(refArray[array [i]]);
 }
 s.append("]");
 return s.toString();
}


#ifdef TEST
 private static long seed = System.currentTimeMillis();
 private static java.util.Random r = new java.util.Random(seed);

 private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
  return (KEY_TYPE)(r.nextInt());
```

```
#elif KEYS_PRIMITIVE
  return r.NEXT_KEY();
#elif KEY_CLASS_Object
  return Integer.toBinaryString(r.nextInt());
#else
  return new java.io.Serializable() {};
#endif
 }

 private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
 private static java.text.FieldPosition p = new java.text.FieldPosition(0);

 private
 static String format(double d) {
  StringBuffer s = new StringBuffer();
  return format.format(d, s, p).toString();
 }

 private static void speedTest(int n, boolean comp) {
  int i, j, s;
  ARRAY_INDIRECT_PRIORITY_QUEUE[] m = new ARRAY_INDIRECT_PRIORITY_QUEUE[100000];
  HEAP_INDIRECT_PRIORITY_QUEUE[] t = new HEAP_INDIRECT_PRIORITY_QUEUE[m.length];
  KEY_TYPE k[] = new KEY_TYPE[n];
  KEY_TYPE nk[] = new KEY_TYPE[m.length];
  long ms;

  for(i = 0; i < n; i++) k[i] = genKey();
  for(i = 0; i < m.length; i++) nk[i] = genKey();

  double totEnq = 0, totDeq = 0, totChange = 0, d;

  for(i = 0; i < m.length; i++) {
   t[i] = new HEAP_INDIRECT_PRIORITY_QUEUE(k);
   m[i] = new ARRAY_INDIRECT_PRIORITY_QUEUE(k);
  }

  if (comp) {
   for(j = 0; j < 20; j++) {

    for(i = 0; i < m.length; i++) t[i].clear();

    ms = System.currentTimeMillis();
    s = m.length;
    while(s-- != 0) { i = n; while(i-- != 0) t[s].enqueue(i); }
    d = System.currentTimeMillis() - ms;
    if (j > 2) totEnq
 += d;
    System.out.print("Enqueue: " + format(m.length * n/d) +" K/s ");
```

```
    ms = System.currentTimeMillis();
    s = m.length;
    while(s-- != 0) { i = n; while(i-- != 0) { k[t[s].first()] = nk[i]; t[s].changed(); } }
    d = System.currentTimeMillis() - ms;
    if (j > 2) totChange += d;
    System.out.print("Change: " + format(m.length * n/d) +" K/s ");

    ms = System.currentTimeMillis();
    s = m.length;
    while(s-- != 0) { i = n; while(i-- != 0) t[s].dequeue(); }
    d = System.currentTimeMillis() - ms;
    if (j > 2) totDeq += d;
    System.out.print("Dequeue: " + format(m.length * n/d) +" K/s ");

    System.out.println();
   }

   System.out.println();
   System.out.println("Heap:  Enqueue: " + format(m.length * (j-3)*n/totEnq) + " K/s Dequeue: " + format(m.length *
(j-3)*n/totDeq)  + " K/s Change: " + format(m.length * (j-3)*n/totChange) + " K/s");

   System.out.println();

   totEnq = totChange = totDeq = 0;

  }

  for(j = 0; j < 20; j++) {

   for(i = 0; i <
m.length; i++) m[i].clear();

   ms = System.currentTimeMillis();
   s = m.length;
   while(s-- != 0) { i = n; while(i-- != 0) m[s].enqueue(i); }
   d = System.currentTimeMillis() - ms;
   if (j > 2) totEnq += d;
   System.out.print("Enqueue: " + format(m.length * n/d) +" K/s ");

   ms = System.currentTimeMillis();
   s = m.length;
   while(s-- != 0) { i = n; while(i-- != 0) { k[m[s].first()] = nk[i]; m[s].changed(); } }
   d = System.currentTimeMillis() - ms;
   if (j > 2) totChange += d;
   System.out.print("Change: " + format(m.length * n/d) +" K/s ");

   ms = System.currentTimeMillis();
```

```
  s = m.length;
  while(s-- != 0) { i = n; while(i-- != 0) m[s].dequeue(); }
  d = System.currentTimeMillis() - ms;
  if (j > 2) totDeq += d;
  System.out.print("Dequeue: " + format(m.length * n/d) +" K/s ");

  System.out.println();
 }


 System.out.println();
 System.out.println("Array:  Enqueue: " + format(m.length * (j-3)*n/totEnq) + " K/s Dequeue: " + format(m.length *
(j-3)*n/totDeq)
 + " K/s Change: " + format(m.length * (j-3)*n/totChange) + " K/s");

 System.out.println();
}


private static void fatal(String msg) {
 System.out.println(msg);
 System.exit(1);
}

private static void ensure(boolean cond, String msg) {
 if (cond) return;
 fatal(msg);
}

private static boolean heapEqual(int[] a, int[] b, int sizea, int sizeb) {
 if (sizea != sizeb) return false;
 KEY_TYPE[] aa = new KEY_TYPE[sizea];
 KEY_TYPE[] bb = new KEY_TYPE[sizea];
 for(int i = 0; i < sizea; i++) {
  aa[i] = ref[a[i]];
  bb[i] = ref[b[i]];
 }
 java.util.Arrays.sort(aa);
 java.util.Arrays.sort(bb);
 while(sizea-- != 0) if (!KEY_EQUALS(aa[sizea], bb[sizea])) return false;
 return true;
}

private static KEY_TYPE[] ref;

protected static void runTest(int n) {
 long ms;
 Exception mThrowsIllegal, tThrowsIllegal, mThrowsOutOfBounds, tThrowsOutOfBounds, mThrowsNoElement,
```

```
tThrowsNoElement;
  int rm = 0, rt = 0;

  ref = new KEY_TYPE[n];

  for(int i = 0; i < n; i++) ref[i] =
 genKey();

  ARRAY_INDIRECT_PRIORITY_QUEUE m = new ARRAY_INDIRECT_PRIORITY_QUEUE(ref);
  HEAP_INDIRECT_PRIORITY_QUEUE t = new HEAP_INDIRECT_PRIORITY_QUEUE(ref);

  /* We add pairs to t. */
  for(int i = 0; i < n / 2;  i++) {
   t.enqueue(i);
   m.enqueue(i);
  }

  ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and t differ after creation (" + m + ", "
+ t + ")");

  /* Now we add and remove random data in m and t, checking that the result is the same. */

  for(int i=0; i<2*n;  i++) {
   if (r.nextDouble() < 0.01) {
    t.clear();
    m.clear();
    for(int j = 0; j < n / 2;  j++) {
     t.enqueue(j);
     m.enqueue(j);
    }
   }

   int T = r.nextInt(2 * n);

   mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

   try {
    t.enqueue(T);
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { tThrowsIllegal = e; }

   if (tThrowsIllegal
 == null) { // To skip duplicates
    try {
     m.enqueue(T);
    }
```

```
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
    catch (IllegalArgumentException e) { mThrowsIllegal = e; }
   }

   mThrowsIllegal = tThrowsIllegal = null; // To skip duplicates

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): enqueue()
divergence in IndexOutOfBoundsException for " + T + " (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds
+ ")");
   ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): enqueue() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");

   ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and t differ after enqueue (" + m + ", "
+ t + ")");

   if (m.size() != 0) {
    ensure(KEY_EQUALS(ref[m.first()], ref[t.first()]), "Error (" + seed + "): m and t differ in first element after
enqueue (" + m.first()
 + "->" + ref[m.first()] + ", " + t.first() + "->" + ref[t.first()] + ")");
   }

   mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

   try {
    rm = m.dequeue();
    while(! m.isEmpty() && KEY_EQUALS(ref[m.first()], ref[rm])) m.dequeue();
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { mThrowsIllegal = e; }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = e; }

   try {
    rt = t.dequeue();
    while(! t.isEmpty() && KEY_EQUALS(ref[t.first()], ref[rt])) t.dequeue();
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { tThrowsIllegal = e; }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): dequeue()
divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds
 + ", " + tThrowsOutOfBounds + ")");
   ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): dequeue() divergence in
IllegalArgumentException  (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
   ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): dequeue() divergence
in java.util.NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
   if (mThrowsOutOfBounds == null) ensure(KEY_EQUALS(ref[rt], ref[rm]), "Error (" + seed + "): divergence in
```

```
dequeue() between m and t (" + rm + "->" + ref[rm] + ", " + rt + "->" + ref[rt] + ")");


   ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and t differ after dequeue (" + m + ", "
+ t + ")");

   if (m.size() != 0) {
    ensure(KEY_EQUALS(ref[m.first()], ref[t.first()]), "Error (" + seed + "): m and t differ in first element after
dequeue (" + m.first() + "->" + ref[m.first()] + ", " + t.first() + "->" + ref[t.first()]
 + ")");
   }

   mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;


   int pos = r.nextInt(n * 2);

   try {
    m.remove(pos);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { mThrowsIllegal = e; }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = e; }

   try {
    t.remove(pos);
   }
   catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { tThrowsIllegal = e; }
   catch (java.util.NoSuchElementException e) { tThrowsNoElement = e; }

   ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): remove(int)
divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");
   ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): remove(int) divergence in
IllegalArgumentException  (" + mThrowsIllegal
 + ", " + tThrowsIllegal + ")");
   ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): remove(int)
divergence in java.util.NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");

   ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and t differ after remove(int) (" + m +
", " + t + ")");

   if (m.size() != 0) {
    ensure(KEY_EQUALS(ref[m.first()], ref[t.first()]), "Error (" + seed + "): m and t differ in first element after
remove(int) (" + m.first() + "->" + ref[m.first()] + ", " + t.first() + "->" + ref[t.first()] + ")");
   }
```

```
  mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

  pos = r.nextInt(n);

  try {
   t.changed(pos);
  }
  catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
  catch (IllegalArgumentException e) { tThrowsIllegal = e; }
  catch (java.util.NoSuchElementException e)
 { tThrowsNoElement = e; }

  if (tThrowsIllegal == null) {
   try {
    m.changed(pos);
   }
   catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
   catch (IllegalArgumentException e) { mThrowsIllegal = e; }
   catch (java.util.NoSuchElementException e) { mThrowsNoElement = e; }
  }

  ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): change(int)
divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");
  //ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): change(int) divergence in
IllegalArgumentException  (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
  ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): change(int)
divergence in java.util.NoSuchElementException  (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");

  ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and
 t differ after change(int) (" + m + ", " + t + ")");

  if (m.size() != 0) {
   ensure(KEY_EQUALS(ref[m.first()], ref[t.first()]), "Error (" + seed + "): m and t differ in first element after
change(int) (" + m.first() + "->" + ref[m.first()] + ", " + t.first() + "->" + ref[t.first()] + ")");
  }

  int[] temp = (int[])t.heap.clone();
  java.util.Arrays.sort(temp, 0, t.size()); // To scramble a bit
  m = new ARRAY_INDIRECT_PRIORITY_QUEUE(m.refArray, temp, t.size());

  ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and t differ after wrap (" + m + ", " + t
+ ")");

  if (m.size() != 0) {
   ensure(KEY_EQUALS(ref[m.first()], ref[t.first()]), "Error (" + seed + "): m and t differ in first element after wrap ("
+ m.first() + "->" + ref[m.first()] + ", " + t.first() + "->" + ref[t.first()] + ")");
```

```
  }

  if (m.size() != 0 && ((new it.unimi.dsi.fastutil.ints.IntOpenHashSet(m.array, 0, m.size)).size() == m.size())) {

   int first = m.first();
   ref[first]
 = genKey();

   //System.err.println("Pre-change m: " +m);
   //System.err.println("Pre-change t: " +t);
   m.changed();
   t.changed(first);

   //System.err.println("Post-change m: " +m);
   //System.err.println("Post-change t: " +t);

   ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and t differ after change (" + m + ", " +
t + ")");

   if (m.size() != 0) {
    ensure(KEY_EQUALS(ref[m.first()], ref[t.first()]), "Error (" + seed + "): m and t differ in first element after change
(" + m.first() + "->" + ref[m.first()] + ", " + t.first() + "->" + ref[t.first()] + ")");
   }
  }
 }


 /* Now we check that m actually holds the same data. */

 m.clear();
 ensure(m.isEmpty(), "Error (" + seed + "): m is not empty after clear()");

 System.out.println("Test OK");
}



public static void main(String args[]) {
 int n  = Integer.parseInt(args[1]);
 if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));


 try
 {
  if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
  else if ("test".equals(args[0])) runTest(n);
 } catch(Throwable e) {
  e.printStackTrace(System.err);
```

```
  System.err.println("seed: " + seed);
 }
}

#endif
}
```

Found in path(s):
* /opt/cola/permits/2100345329_1721125147.3484588/0/cloudburstmc-fastutil-8-4-0-0-g6e390a2-tar-gz/CloudburstMC-fastutil-6e390a2/drv/ArrayIndirectPriorityQueue.drv

# 1.247 concurrent-futures 1.0.0

## 1.247.1 Available under license :

Apache-2.0

# 1.248 kotlin-tooling-metadata 1.7.10

## 1.248.1 Available under license :

Apache-2.0

# 1.249 kotlin-tooling-core 1.7.10

## 1.249.1 Available under license :

Apache-2.0

# 1.250 retrofit2-kotlin-coroutines-adapter 0.9.2

## 1.250.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.251 openapi-generator 5.2.1

## 1.251.1 Available under license :

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2018, Daniel Stenberg, daniel@haxx.se, and many contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization of the copyright holder.
QHttpEngine

The MIT License (MIT)

Copyright (c) 2015 Nathan Osman

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
{
 "name": "graphql-server",
 "version": "1.0.0",
 "description": "{{#appDescription}} {{{appDescription}}} {{/appDescription}}",
 "main": "index.js",
 "scripts": {
   "start": "node start.js",
   "test": "echo \"Error: no test specified\" && exit 1"
 },
 "keywords": [],
 "author": "",
 "license": "ISC",
 "dependencies": {
   "express": "latest",
   "graphql": "latest",
   "apollo-server-express": "latest",
   "graphql-combine": "latest",
   "lodash": "latest",
   "path": "latest"
 },
 "devDependencies": {
   "babel-register": "latest",
   "babel-preset-env": "latest"
 }
}
{
 "name": "{{#npmName}}{{{npmName}}}{{/npmName}}{{^npmName}}typescript-aurelia-api{{/npmName}}",
 "version": "{{#npmVersion}}{{{npmVersion}}}{{/npmVersion}}{{^npmVersion}}0.0.0{{/npmVersion}}",
 "license": "Unlicense",
 "main": "./dist/Api.js",
 "browser": "./dist/Api.js",
 "typings": "./dist/Api.d.ts",
 "dependencies": {
   {{^supportsES6}}"core-js": "^2.4.0",
   {{/supportsES6}}"isomorphic-fetch": "^2.2.1",
   "aurelia-framework": "^1.3.1",
   "aurelia-http-client": "^1.3.0"
 },
 "scripts": {
   "build": "tsc --outDir dist/",
   "prepare": "npm run build",
   "test": "tslint Api.ts"
 },
 "devDependencies": {
   "tslint": "^3.15.1",
   "typescript": "^2.4 || ^3.0"
```

}
}
Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF
OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
=begin comment

{{#appName}}
{{{appName}}}

{{/appName}}
{{#appDescription}}
{{{appDescription}}}

{{/appDescription}}
{{#version}}The version of the OpenAPI document: {{version}}{{/version}}
{{#infoEmail}}Contact: {{{infoEmail}}}{{/infoEmail}}
Generated by: https://openapi-generator.tech

=end comment

=cut
This software is licensed under the Apache 2 license, quoted below.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this project except in compliance with
the License. You may obtain a copy of the License at https://www.apache.org/licenses/LICENSE-2.0.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific

language governing permissions and limitations under the License.

# 1.252 dokka-core 1.5.30

## 1.252.1 Available under license :

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations,
or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems
that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants
to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work
or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution
of Your modifications, or
for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental,
or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted
against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{}"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright 2014-2019 JetBrains s.r.o. and Dokka project contributors.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.253 apksig 7.3.0

## 1.253.1 Available under license :

Copyright (c) 2016, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within
the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible
for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed
in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.254 core-common 2.0.0

## 1.254.1 Available under license :

Apache-2.0

# 1.255 incap 0.2

## 1.255.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright © 2018 Thomas Broyer
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345133_1721125078.5978882/0/incap-0-2-sources-jar/net/ltgt/gradle/incap/IncrementalAnnotationProcessor.java
* /opt/cola/permits/2100345133_1721125078.5978882/0/incap-0-2-sources-

jar/net/ltgt/gradle/incap/IncrementalAnnotationProcessorType.java

# 1.256 crash 30.3.0

## 1.256.1 Available under license :

Copyright (c) 2005-2016, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form
of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.257 kotlin-stdlib-jdk7 1.7.10

## 1.257.1 Available under license :

Apache-2.0

# 1.258 builder-test-api 7.3.0

## 1.258.1 Available under license :

Copyright (c) 2005-2017, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited
to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely

responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.259 httpmime 4.5.6

## 1.259.1 Available under license :

Apache HttpClient Mime
Copyright 1999-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.260 kotlin-stdlib-jdk8 1.7.10

## 1.260.1 Available under license :

Apache-2.0

# 1.261 kotlin-stdlib-common 1.7.0

## 1.261.1 Available under license :

Apache-2.0

# 1.262 kotlin-stdlib-common 1.8.0

## 1.262.1 Available under license :

Apache-2.0

# 1.263 kotlin-daemon-embeddable 1.5.31

## 1.263.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2016 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and

* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/report/DaemonMessageReporter.kt
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/report/CompileServicesFacadeMessageCollector.kt
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/RemoteInputStreamClient.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/KotlinRemoteReplService.kt
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/report/getICReporter.kt
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/RemoteReplStateFacadeImpl.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2015 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/RemoteCompilationCanceledStatusClient.kt
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/KotlinCompileDaemon.kt
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/RemoteIncrementalCompilationComponentsClient.kt
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/RemoteIncrementalCacheClient.kt
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/LazyClasspathWatcher.kt
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/RemoteOutputStreamClient.kt
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/CompileServiceImpl.kt
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/RemoteLookupTrackerClient.kt
* /opt/cola/permits/2100344698_1721124939.7705479/0/kotlin-daemon-embeddable-1-5-31-sources-1-jar/org/jetbrains/kotlin/daemon/experimental/RemoteCompilationCanceledStatusClient.kt

# 1.264 kotlin-stdlib-jdk7 1.7.0

## 1.264.1 Available under license :

Apache-2.0

# 1.265 lifecycle-common 2.5.1

## 1.265.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2018 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/LifecycleEventObserver.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/GenericLifecycleObserver.java
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/GeneratedAdapter.java
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/FullLifecycleObserver.java
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/MethodCallsLogger.java
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/DefaultLifecycleObserver.java
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/CompositeGeneratedAdaptersObserver.java
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/Lifecycle.java
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/Lifecycling.java
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/OnLifecycleEvent.java

* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/FullLifecycleObserverAdapter.java
*
 /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/ReflectiveGenericLifecycleObserver.java
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/SingleGeneratedAdapterObserver.java
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/LifecycleOwner.java
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/ClassesInfoCache.java
* /opt/cola/permits/2100345008_1721146176.8068082/0/lifecycle-common-2-5-1-sources-jar/androidx/lifecycle/LifecycleObserver.java

# 1.266 kotlin-stdlib-jdk8 1.5.0

## 1.266.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345186_1721125100.2508137/0/kotlin-stdlib-jdk8-1-5-0-sources-jar/kotlin/internal/jdk8/JDK8PlatformImplementations.kt
* /opt/cola/permits/2100345186_1721125100.2508137/0/kotlin-stdlib-jdk8-1-5-0-sources-jar/kotlin/text/RegexExtensions.kt
* /opt/cola/permits/2100345186_1721125100.2508137/0/kotlin-stdlib-jdk8-1-5-0-sources-jar/kotlin/streams/Streams.kt
*
 /opt/cola/permits/2100345186_1721125100.2508137/0/kotlin-stdlib-jdk8-1-5-0-sources-jar/kotlin/collections/Collections.kt

# 1.267 kotlin-gradle-plugin-api 1.7.10-gradle70

## 1.267.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2014 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345052_1721144426.4193347/0/kotlin-gradle-plugin-api-1-7-10-gradle70-sources-jar/org/jetbrains/kotlin/gradle/plugin/KotlinGradleSubplugin.kt

# 1.268 kotlin-compiler-embeddable 1.5.31

## 1.268.1 Available under license :

Apache-2.0

# 1.269 kotlin-stdlib-jdk7 1.6.10

## 1.269.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
```

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345236_1721125186.4661076/0/kotlin-stdlib-jdk7-1-6-10-sources-2-jar/kotlin/AutoCloseable.kt

# 1.270 gradle-download-task 4.1.1

## 1.270.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.271 constraintlayout-solver 1.1.3

## 1.271.1 Available under license :

Apache-2.0

# 1.272 kotlin-stdlib 1.7.10

## 1.272.1 Available under license :

Apache-2.0

# 1.273 kotlin-scripting-compiler-embeddable 1.5.31

## 1.273.1 Available under license :

Apache-2.0

# 1.274 builder 7.3.0

## 1.274.1 Available under license :

Copyright (c) 2005-2012, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.275 manifest-merger 30.3.0

## 1.275.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2015 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/Merger.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 The Android Open Source Project
*
```

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/ManifestProvider.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/ConvertibleName.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/PreValidator.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/AttributeModel.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/Selector.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/XmlLoader.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/PostValidator.java

* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/KeyResolver.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/NodeOperationType.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/ElementsTrimmer.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/ToolsInstructionsCleaner.java
*
 /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/XmlElement.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/XmlDocument.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/PlaceholderHandler.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/Actions.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/OrphanXmlElement.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/MergingReport.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/XmlNode.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/AttributeOperationType.java
*
 /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/OtherOperationType.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/PlaceholderEncoder.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/ActionRecorder.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/XmlAttribute.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/ManifestMerger2.java
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/MergeType.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2014 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
```

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/** attribution (contained in manifest) */

Found in path(s):
* /opt/cola/permits/2100344651_1721144916.5769482/0/manifest-merger-30-3-0-sources-jar/com/android/manifmerger/ManifestModel.java

# 1.276 kotlin-reflect 1.5.31

## 1.276.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2016 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/AbstractDeserializedPackageFragmentProvider.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/AbstractClassTypeConstructor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/GivenFunctionsMemberScope.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/OverridingStrategy.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/context.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-

jar/kotlin/reflect/jvm/internal/impl/load/kotlin/methodSignatureBuildingUtils.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/full/KClassifiers.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/kotlin/DeserializationComponentsForJava.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/types/JavaTypeResolver.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/constants/IntegerValueTypeConstructor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaTypeParameterDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/KClassifierImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/kotlin/JavaFlexibleTypeDeserializer.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/IntersectionTypeConstructor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/calls/inference/CapturedTypeConstructor.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/ValueDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/constants/CompileTimeConstant.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/utils/SmartList.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/DescriptorEquivalenceForOverrides.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/AbstractTypeConstructor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/checker/KotlinTypeChecker.java
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/AnnotationsImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/renderer/DescriptorRendererOptionsImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/KotlinReflectionInternalError.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/ClassifierDescriptorWithTypeParameters.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/Annotations.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/JvmPackageScope.kt
*

/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/ClassifierDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/renderer/ClassifierNamePolicy.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/TypeIntersectionScope.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/SuperCallReceiverValue.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/FunctionPlaceholders.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/checker/TypeCheckingProcedure.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/KTypesJvm.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/types/RawType.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/CapturedTypeApproximation.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/utils/DFS.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/FlexibleTypeDeserializer.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/ErrorUtils.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/dynamicTypes.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/EmptyPackageFragmentDesciptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/flexibleTypes.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/components/JavaPropertyInitializerEvaluator.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/checker/utils.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/full/KTypes.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/TypeAliasDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/AnnotationDeserializer.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/descriptors/DeserializedTypeParameterDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/TypeCapabilities.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-

jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaAnnotationDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/MetadataPackageFragmentProvider.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/TypeSubstitutor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/full/exceptions.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/TypeSubstitution.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/checker/ClassicTypeCheckerContext.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/ClassConstructorDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/TypeProjectionBase.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/full/KClasses.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/typeEnhancement/typeEnhancement.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/RawType.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/KTypeParameterImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/ErasedOverridabilityCondition.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/context.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/AbstractTypeParameterDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/checker/KotlinTypeCheckerImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/AbstractTypeAliasDescriptor.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/AbstractClassDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/utils/functions.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at

*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/ModalityUtils.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/RuntimeModuleData.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaStaticClassScope.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/findClassInModule.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/descriptors/JavaClassConstructorDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/util/modifierChecks.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/BuiltInAnnotationDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/KotlinType.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/InvalidModuleException.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/checker/NewKotlinTypeChecker.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaConstructor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/ReflectKotlinClassFinder.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaTypeParameter.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/utils.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaType.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaAnnotationOwner.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/DescriptorVisibilities.java

* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaAnnotation.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/JavaClassesTracker.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/AnnotationTypeQualifierResolver.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/utils/numbers.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaAnnotationArguments.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/TypeWithEnhancement.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/RuntimeSourceElementFactory.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/deserialization/AdditionalClassPartsProvider.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/PropertyDescriptorImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/container/DefaultImplementation.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/ClassConstructorDescriptorImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/ConstructorDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/DescriptorDerivedFromTypeAlias.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaArrayType.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/OverridingUtil.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/deserialization/ClassDescriptorFactory.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/checker/NewCapturedType.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/ContractDeserializer.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/builtins/BuiltInsLoader.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaWildcardType.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaPackageScope.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaScope.kt
*

/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/SyntheticScopes.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/DeserializedPackageFragmentImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaClassifierType.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaValueParameter.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaPrimitiveType.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/checker/IntersectionType.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaClass.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaMethod.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaPackage.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/ErrorType.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/SpecialTypes.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/builtins/BuiltInsPackageFragment.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/ReflectJavaClassFinder.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/Substitutable.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaClassMemberScope.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/typeEnhancement/signatureEnhancement.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/TypeAliasConstructorDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/KotlinTypeFactory.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/deserialization/PlatformDependentDeclarationFilter.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaMember.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/TypeUtils.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/ReflectKotlinClass.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-

jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/reflectClassUtil.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/full/KProperties.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2015 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
// A seemingly obvious way to come about this case would be to declare a special exception class, but the problem is that
```

Found in path(s):
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/storage/LockBasedStorageManager.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaField.kt

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2015 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/components/JavaResolverCache.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/resolvers.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/AnnotationDescriptorImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/TypeProjectionImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/Named.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/descriptors/DeserializedPackageMemberScope.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaPackageFragment.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/checker/TypeCheckingProcedureCallbacks.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/KTypeImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/AnnotationAndConstantLoaderImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/PropertyDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/utils/capitalizeDecapitalize.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/KDeclarationContainerImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-

jar/kotlin/reflect/jvm/internal/impl/load/java/descriptors/JavaClassDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/pcollections/ConsPStack.java
*
 /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/SubpackagesScope.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/util/ModuleVisibilityHelper.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/SubstitutingScope.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/load/kotlin/AbstractBinaryClassAnnotationAndConstantLoader.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/descriptors/PackageViewDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/ReceiverParameterDescriptorImpl.java
*
 /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/DeclarationDescriptorNonRootImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/ResolutionScope.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/ImplicitReceiver.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/resolve/constants/constantValues.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/incremental/utils.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/ClassDescriptorBase.java
*
 /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/KotlinRetention.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/EmptyContainerForLocal.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/descriptors/ModuleDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/kClassCache.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/descriptors/PackageFragmentProviderImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/descriptors/ReceiverParameterDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/load/java/descriptors/JavaPropertyDescriptor.java
*
 /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-
jar/kotlin/reflect/jvm/internal/impl/descriptors/TypeParameterDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-

jar/kotlin/reflect/jvm/internal/impl/descriptors/ParameterDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/sources/JavaSourceElementFactory.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/ReflectionFactoryImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/moduleByClassLoader.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/LazyJavaPackageFragmentProvider.kt
*
 /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/LazyPackageViewDescriptorImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/AbstractReceiverValue.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/ClassOrPackageFragmentDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/storage/NoLock.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/JavaDescriptorUtil.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/PackageFragmentDescriptorImpl.kt
*
 /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/pcollections/HashPMap.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/kotlin/DeserializedDescriptorResolver.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/DeclarationDescriptorVisitorEmptyBodies.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/utils/Printer.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/components/TypeUsage.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/kotlin/BinaryClassAnnotationAndConstantLoaderImpl.kt
*
 /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/DeclarationDescriptorVisitor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/KCallablesJvm.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/AbstractLazyTypeParameterDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/KProperty0Impl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/MemberScopeImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/KParameterImpl.kt

* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/PropertySetterDescriptor.java
*
 /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/ReceiverValue.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/DescriptorVisibility.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/ValueParameterDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/JavaIncompatibilityRulesOverridabilityCondition.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/components/DescriptorResolverUtils.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/builtins/DefaultBuiltIns.kt
*
 /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/ExtensionReceiver.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/CallableMemberDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/AnnotationArgumentVisitor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/FieldOverridabilityCondition.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/KProperty2Impl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/specialBuiltinMembers.kt
*
 /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/kotlin/header/ReadKotlinClassHeaderAnnotationVisitor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/VariableAccessorDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/PropertyAccessorDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/ConstUtil.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/annotationUtil.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/utils/SmartSet.kt
*
 /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/DisjointKeysUnionTypeSubstitution.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/util/scopeUtils.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/ExternalOverridabilityCondition.java

* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/ErrorReporter.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/descriptors/util.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/PackageFragmentProvider.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/storage/StorageManager.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/DeclarationDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/LazyClassReceiverParameterDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/KProperty1Impl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/utils/collections.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/AnnotatedImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/components/JavaAnnotationMapper.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/CastImplicitClassReceiver.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/typeParameterUtils.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/descriptors/DeserializedAnnotations.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/AbstractScopeAdapter.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/StaticScopeForKotlinEnum.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/InnerClassesScopeWrapper.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/CallableDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/AnnotatedCallableKind.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/CompositePackageFragmentProvider.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/DeclarationDescriptorWithVisibility.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/ModuleClassResolver.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/descriptors/DeserializedMemberScope.kt
*

/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/PropertyGetterDescriptorImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/utils/strings.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/Receiver.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/TypeProjection.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/SimpleFunctionDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/KClassesJvm.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/DescriptorFactory.java
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/ReflectAnnotationSource.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/utils/exceptionUtils.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/KClassImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/descriptors/JavaCallableMemberDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/MemberComparator.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/ValueParameterDescriptorImpl.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/MemberDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/ClassDeserializer.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/KPackageImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/PropertyAccessorDescriptorImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/LazyScopeAdapter.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/AbstractReceiverParameterDescriptor.java
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/ReflectProperties.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/DeclaredMemberIndex.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/ProtoContainer.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-

jar/kotlin/reflect/jvm/internal/impl/types/CompositionTypeSubstitution.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/VariableDescriptorWithAccessors.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/VariableDescriptorWithInitializerImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/JavaDescriptorVisibilities.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/AnnotationWithTarget.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/checker/TypeCheckerProcedureCallbacksImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/annotations/AnnotationDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/FunctionWithAllInvokes.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/ReflectJvmMapping.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/components/SignaturePropagator.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/RuntimeTypeMapper.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/ReflectionObjectRenderer.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/StarProjectionImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/DescriptorSubstitutor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/FakePureImplementationsProvider.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/reflectLambda.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/components/RuntimeErrorReporter.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/utils/coreLib.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/util.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/builtins/functions/FunctionInvokeDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/builtins/functions/BuiltInFictitiousFunctionClassFactory.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/MemberScope.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/pcollections/MapEntry.java

*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/SimpleFunctionDescriptorImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/ScriptDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaElement.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/SupertypeLoopChecker.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/TypeParameterDescriptorImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/pcollections/IntTree.java
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/utils/WrappedValues.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/TransientReceiver.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/LocalClassifierTypeSettings.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/PropertyGetterDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/receivers/ImplicitClassReceiver.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/TypeConstructor.java
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/LazyJavaAnnotations.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/KFunctionImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/DeclarationDescriptorImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/VisibilityUtil.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/builtins/functions/FunctionClassScope.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/runtime/structure/ReflectJavaModifierListOwner.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/scopes/ChainedMemberScope.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/types/ClassTypeConstructorImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/DeserializedClassDataFinder.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/DeclarationDescriptorWithSource.java

* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/PackageFragmentDescriptor.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/lazy/descriptors/LazyJavaStaticScope.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/VariableDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/pcollections/IntTreePMap.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/kotlin/KotlinJvmBinaryPackageSourceElement.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/jvm/JavaDescriptorResolver.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/AnnotationAndConstantLoader.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/ModuleDescriptorImpl.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/storage/storage.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/DeclarationDescriptorNonRoot.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/storage/ObservableStorageManager.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/PropertySetterDescriptorImpl.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/overridingUtils.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/resolve/constants/ConstantValueFactory.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/kotlin/JavaClassDataFinder.kt
*
/opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/load/java/descriptors/JavaMethodDescriptor.java
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/serialization/deserialization/DeserializedPackageFragment.kt
* /opt/cola/permits/2100344724_1721124942.9659932/0/kotlin-reflect-1-5-31-sources-1-jar/kotlin/reflect/jvm/internal/impl/descriptors/impl/VariableDescriptorImpl.java

# 1.277 kotlinx-coroutines-core-jvm 1.5.1

## 1.277.1 Available under license :

Apache-2.0

# 1.278 sdk-common 30.3.0

## 1.278.1 Available under license :

Copyright (c) 2005-2008, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.279 jackson-dataformat-xml 2.12.4

## 1.279.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

# 1.280 android-test-plugin-host-coverage-proto 30.3.0

## 1.280.1 Available under license :

Copyright (c) 2005-2021, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited
to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.281 kotlinx-coroutines-jdk 1.5.0

## 1.281.1 Available under license :

Apache-2.0

# 1.282 android-test-plugin-host-retention-proto 30.3.0

## 1.282.1 Available under license :

Copyright (c) 2005-2008, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited
to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated

within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.283 swagger-parser 2.0.26

## 1.283.1 Available under license :

Apache-2.0

# 1.284 kotlinx-coroutines-jdk 1.6.1

## 1.284.1 Available under license :

Apache-2.0

# 1.285 openapi-generator-core 5.2.1

## 1.285.1 Available under license :

Apache-2.0

# 1.286 android-device-provider-gradle-proto 30.3.0

## 1.286.1 Available under license :

Copyright (c) 2005-2021, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited
to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.287 kotlin-project-model 1.5.31

## 1.287.1 Available under license :

Apache-2.0

# 1.288 dagger-spi 2.44

## 1.288.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/package-info.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/DiagnosticReporter.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/internal/codegen/extension/DaggerGraphs.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/model/ComponentPath.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Dagger Authors.
*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/internal/codegen/extension/Optionals.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/model/BindingGraph.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2022 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/internal/codegen/xprocessing/XProcessingEnvs.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/internal/codegen/xprocessing/JavaPoetExt.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/model/Key.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/model/DependencyRequest.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/internal/codegen/extension/DaggerCollectors.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/RequestKind.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/Key.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/internal/codegen/xprocessing/XAnnotations.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/internal/codegen/xprocessing/XAnnotationValues.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/DaggerType.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/ComponentPath.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/DaggerAnnotation.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/internal/codegen/xprocessing/XMethodElements.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/Scope.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/DaggerProcessingEnv.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/DependencyRequest.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/BindingGraphPlugin.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/internal/codegen/xprocessing/XTypes.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/MoreAnnotationMirrors.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/BindingGraph.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/DiagnosticReporter.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/DaggerElement.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/internal/codegen/xprocessing/MethodSpecs.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/internal/codegen/xprocessing/XTypeElements.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/BindingKind.java

* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/package-info.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/DaggerExecutableElement.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/CompilerEnvironment.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/DaggerTypeElement.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/internal/codegen/xprocessing/XElements.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/model/Binding.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/spi/BindingGraphPlugin.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/model/BindingKind.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/model/package-info.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/model/Binding.java
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/model/RequestKind.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/internal/codegen/extension/DaggerStreams.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344827_1721124804.4348803/0/dagger-spi-2-44-sources-jar/dagger/model/Scope.java

# 1.289 kotlin-scripting-common 1.5.31

## 1.289.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2018 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344745_1721125096.6715739/0/kotlin-scripting-common-1-5-31-sources-1-jar/kotlin/script/experimental/host/BasicScriptingHost.kt

# 1.290 dokka-gradle-plugin 1.5.30

## 1.290.1 Available under license :

Apache-2.0

# 1.291 baselibrary 7.3.0

## 1.291.1 Available under license :

Copyright (c) 2005-2017, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited
to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely

responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.292 kotlin-gradle-plugin 1.5.31

## 1.292.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/dsl/optionsUtil.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/tasks/CacheableTasks.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/dsl/KotlinJsDceOptionsImpl.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/utils/compatibiltiy.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/tasks/KotlinJsDce.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/internal/kapt/KaptGenerateStubsTask.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/internal/CompilerArgumentAware.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/dsl/KotlinMultiplatformCommonOptionsImpl.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/compilerRunner/reportUtils.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/internal/subpluginUtils.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/internal/CompilerArgumentsGradleInput.kt
No license file was found, but licenses were detected in source scan.

var license: String? = null

Found in path(s):
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/targets/native/cocoapods/CocoapodsExtension.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2015 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/utils/Printer.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2016 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/tasks/KotlinCompileCommon.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/plugin/gradleUtils.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/plugin/KotlinMultiplatformPlugin.kt
*
 /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/dsl/KotlinJvmOptionsImpl.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/dsl/KotlinCompile.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/tasks/TasksProvider.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/plugin/KaptExtension.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/plugin/KotlinPluginWrapper.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/utils/ParsedGradleVersion.kt
*

/opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/dsl/KotlinJsOptionsImpl.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/tasks/jarSearchingUtil.kt
* /opt/cola/permits/2100345372_1721124983.7773685/0/kotlin-gradle-plugin-1-5-31-sources-jar/org/jetbrains/kotlin/gradle/tasks/CompilerPluginOptions.kt

# 1.293 kotlin-compiler-runner 1.5.31

## 1.293.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2015 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344779_1721124799.3203807/0/kotlin-compiler-runner-1-5-31-sources-jar/org/jetbrains/kotlin/compilerRunner/KotlinCompilerRunnerUtils.kt
* /opt/cola/permits/2100344779_1721124799.3203807/0/kotlin-compiler-runner-1-5-31-sources-jar/org/jetbrains/kotlin/compilerRunner/CompilerOutputParser.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2016 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344779_1721124799.3203807/0/kotlin-compiler-runner-1-5-31-sources-jar/org/jetbrains/kotlin/compilerRunner/CompilerEnvironment.kt

# 1.294 dvlib 30.3.0

## 1.294.1 Available under license :

Copyright (c) 2005-2013, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.295 annotation-jvm 1.6.0

## 1.295.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2021 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/GravityInt.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/UiContext.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/DisplayContext.kt
*

/opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/NonUiContext.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2013 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/Nullable.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/NonNull.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2015 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/RequiresPermission.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/FloatRange.kt

* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/WorkerThread.kt
*
 /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/BinderThread.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/Keep.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/UiThread.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/ColorInt.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/CheckResult.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/VisibleForTesting.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/Size.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/CallSuper.kt
*
 /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/TransitionRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/IntRange.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/MainThread.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2021 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/Discouraged.kt

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/ChecksSdkIntAtLeast.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/DoNotInline.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2018 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/RequiresFeature.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2019 The Android Open Source Project
```

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/ContentView.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/InspectableProperty.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2022 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/OpenForTesting.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/DeprecatedSinceApi.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/EmptySuper.kt
*
 /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/RequiresExtension.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-

jar/commonMain/androidx/annotation/ReturnThis.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2017 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/GuardedBy.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/LongDef.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/FontRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/ColorLong.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/NavigationRes.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/AnyThread.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/RestrictTo.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/HalfFloat.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/RequiresApi.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/Dimension.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/DimensionUnit.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/ProductionVisibility.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/Px.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/RestrictTo.jvm.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2014 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/IntDef.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/AnyRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/ColorRes.kt
*

/opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/AttrRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/DimenRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/DrawableRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/StringRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/InterpolatorRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/FractionRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/IntegerRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/MenuRes.kt
*
/opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/XmlRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/StyleRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/StyleableRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/BoolRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/RawRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/AnimRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/ArrayRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/LayoutRes.kt
*
/opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/AnimatorRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/PluralsRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/jvmMain/androidx/annotation/IdRes.kt
* /opt/cola/permits/2100345077_1721145132.091923/0/annotation-jvm-1-6-0-sources-jar/commonMain/androidx/annotation/StringDef.kt

# 1.296 jetifier-core 1.0.0-beta10

## 1.296.1 Available under license :

No license file was found, but licenses were detected in source scan.

# Copyright (C) 2018 The Android Open Source Project
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#  http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License

Found in path(s):
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/default.generated.config
No license file was found, but licenses were detected in source scan.

# Copyright (C) 2017 The Android Open Source Project
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#  http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License

Found in path(s):
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/default.config
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2018 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/pom/DependencyVersions.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/rule/RewriteRulesMap.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/utils/LogLevel.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/pom/PomDependency.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/utils/LogConsumer.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/utils/StdOutLogConsumer.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/TypeRewriter.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/rule/RewriteRule.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/proguard/ProGuardTypesMap.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/type/TypesMap.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/pom/DependencyVersionsMap.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/config/Config.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/pom/PomRewriteRule.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/type/PackageName.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/utils/Log.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/proguard/ProGuardType.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/config/ConfigParser.kt
* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/PackageMap.kt

* /opt/cola/permits/2100344796_1721145976.4260898/0/jetifier-core-1-0-0-beta10-sources-jar/com/android/tools/build/jetifier/core/type/JavaType.kt

# 1.297 collection 1.1.0

## 1.297.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [jianfengye110]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.298 kotlin-stdlib-common 1.7.10

## 1.298.1 Available under license :

Apache-2.0

# 1.299 core-proto 0.0.8-alpha07

## 1.299.1 Available under license :

Apache-2.0

# 1.300 kotlin-reflect 1.7.10

## 1.300.1 Available under license :

Apache-2.0

# 1.301 kotlin-stdlib 1.5.31

## 1.301.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2015 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/CharSequence.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/Any.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/Throwable.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/Nothing.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/annotation/Annotations.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/Iterator.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/Unit.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/Function.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/reflect/KDeclarationContainer.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/Number.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/Char.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/reflect/KParameter.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/Boolean.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/Comparable.kt

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2016 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
```

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/reflect/KVisibility.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/Library.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/internal/InternalAnnotations.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/String.kt
* /opt/cola/permits/2100344754_1721124886.5910757/0/kotlin-stdlib-1-5-31-sources-1-jar/kotlin/ArrayIntrinsics.kt

# 1.302 javapoet 1.13.0

## 1.302.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/LineWrapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 Google, Inc.

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/ClassName.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/ParameterSpec.java
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/JavaFile.java
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/FieldSpec.java
*
 /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/TypeName.java
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/ArrayTypeName.java
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-

jar/com/squareup/javapoet/ParameterizedTypeName.java
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/AnnotationSpec.java
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/CodeWriter.java
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/NameAllocator.java
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/TypeVariableName.java
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/WildcardTypeName.java
*
 /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/TypeSpec.java
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/Util.java
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/CodeBlock.java
* /opt/cola/permits/2100345107_1721124934.1945143/0/javapoet-1-13-0-sources-4-jar/com/squareup/javapoet/MethodSpec.java

# 1.303 kotlin-daemon-client 1.5.31

## 1.303.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2015 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/CompileService.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/FileSystemUtils.kt

* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/NetworkUtils.kt
*
 /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/RemoteOperationsTracer.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/ClientUtils.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/client/RemoteOutputStreamServer.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/DaemonParams.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/client/KotlinCompilerClient.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/RemoteOutputStream.kt
*
 /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/CompilerCallbackServicesFacade.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/client/CompilerCallbackServicesFacadeServer.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/PerfUtils.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2016 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/CompilationOptions.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/client/KotlinRemoteReplCompilerClient.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/RemoteInputStream.kt
*

/opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/CompilerServicesFacadeBase.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/client/RemoteInputStreamServer.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/client/RemoteReplCompilerState.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/CompilationResults.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/IncrementalCompilerServicesFacade.kt
*
/opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/client/NativePlatformUtil.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/JpsCompilerServicesFacade.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/client/BasicCompilerServicesWithResultsFacadeServer.kt
* /opt/cola/permits/2100344924_1721125165.88944/0/kotlin-daemon-client-1-5-31-sources-jar/org/jetbrains/kotlin/daemon/common/ReplStateFacade.kt

# 1.304 dagger 2.28.3

## 1.304.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2015 The Dagger Authors.
*
```

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/multibindings/LongKey.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/multibindings/Multibinds.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/multibindings/ClassKey.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/multibindings/IntKey.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/multibindings/StringKey.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/Beta.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/Subcomponent.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/Provides.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2017 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/MemoizedSentinel.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/MapBuilder.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/SetBuilder.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/multibindings/IntoSet.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/Binds.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/GwtIncompatible.java

*
/opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/BindsOptionalOf.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/BindsInstance.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/multibindings/package-info.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/DoubleCheck.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/Reusable.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/Preconditions.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/multibindings/ElementsIntoSet.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/multibindings/IntoMap.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/ProviderOfLazy.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/MembersInjector.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/Module.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/Lazy.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/ComponentDefinitionType.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/InjectedFieldSignature.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/SingleCheck.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/Component.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/SetFactory.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/AbstractMapFactory.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/MapProviderFactory.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/InstanceFactory.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/DelegateFactory.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/Factory.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/MapFactory.java

* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/MembersInjectors.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/MapKey.java
* /opt/cola/permits/2100345262_1721124812.2288158/0/dagger-2-28-3-sources-jar/dagger/internal/DaggerCollections.java

# 1.305 hilt-compiler 2.44

## 1.305.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2019 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/RootProcessor.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/aggregateddeps/PkgPrivateModuleGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/androidentrypoint/ActivityGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/ComponentDescriptor.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/aggregateddeps/AggregatedDepsGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/androidentrypoint/AndroidEntryPointMetadata.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/Processors.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/aggregateddeps/PkgPrivateEntryPointGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/ProcessorErrorHandler.java

*
/opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/ClassNames.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/ProcessorErrors.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/aggregateddeps/PkgPrivateMetadata.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/androidentrypoint/BroadcastReceiverGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/AnnotationValues.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/Components.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/MoreTypes.java
*
/opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/androidentrypoint/ViewGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/androidentrypoint/Generators.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/ErrorTypeException.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/definecomponent/DefineComponents.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/BaseProcessor.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/androidentrypoint/InjectorEntryPointGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/ComponentGenerator.java
*
/opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/BadInputException.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/androidentrypoint/FragmentGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/ComponentTree.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/aggregateddeps/AggregatedDepsProcessor.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/definecomponent/DefineComponentMetadatas.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/RootType.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/generatesrootinput/GeneratesRootInputProcessor.java
*
/opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/androidentrypoint/ApplicationGenerator.java

* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/AndroidClassNames.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/aggregateddeps/ComponentDependencies.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/definecomponent/DefineComponentBuilderMetadatas.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/generatesrootinput/GeneratesRootInputPropagatedDataGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/Root.java
*
 /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/definecomponent/DefineComponentProcessor.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/generatesrootinput/GeneratesRootInputs.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/androidentrypoint/ServiceGenerator.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2022 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/optionvalues/BooleanValue.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/kotlin/KotlinMetadataUtil.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/kotlin/KotlinMetadataFactory.java
*
 /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/optionvalues/GradleProjectType.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/ElementDescriptors.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-

jar/dagger/hilt/processor/internal/kotlin/KotlinMetadata.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2021 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/aliasof/AliasOfPropagatedDataMetadata.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/ProcessedRootSentinelMetadata.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/ir/MetadataIr.kt
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/uninstallmodules/AggregatedUninstallModulesGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/ComponentTreeDepsMetadata.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/ProcessedRootSentinelGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/ComponentTreeDepsProcessor.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/aggregateddeps/AggregatedDepsMetadata.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/AggregatedRootMetadata.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/AggregatedElements.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/earlyentrypoint/AggregatedEarlyEntryPointGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/earlyentrypoint/EarlyEntryPointProcessor.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/ir/AggregatedRootIrValidator.kt

* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/earlyentrypoint/AggregatedEarlyEntryPointMetadata.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/definecomponent/DefineComponentClassesMetadata.java
*
 /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/ComponentTreeDepsGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/EarlySingletonComponentCreatorGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/ir/ComponentTreeDepsIrCreator.kt
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/uninstallmodules/AggregatedUninstallModulesMetadata.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/AggregatedRootGenerator.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Dagger Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/viewmodel/ViewModelModuleGenerator.kt
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/bindvalue/BindValueProcessor.java
*
 /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/TestComponentDataGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/customtestapplication/CustomTestApplicationProcessor.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/uninstallmodules/UninstallModulesProcessor.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/TestInjectorGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-

jar/dagger/hilt/android/processor/internal/viewmodel/ViewModelValidationPlugin.kt
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/ComponentNames.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/viewmodel/ViewModelProcessor.kt
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/HiltCompilerOptions.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/aliasof/AliasOfs.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/customtestapplication/CustomTestApplicationGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/aliasof/AliasOfProcessor.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/aliasof/AliasOfPropagatedDataGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/RootFileFormatter.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/androidentrypoint/AndroidEntryPointProcessor.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/bindvalue/BindValueMetadata.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/originatingelement/OriginatingElementProcessor.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/RootGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/viewmodel/ViewModelMetadata.kt
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/bindvalue/BindValueGenerator.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/android/processor/internal/customtestapplication/CustomTestApplicationMetadata.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/kotlin/KotlinMetadataUtils.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/TestRootMetadata.java
* /opt/cola/permits/2100344979_1721124916.361296/0/hilt-compiler-2-44-sources-jar/dagger/hilt/processor/internal/root/RootMetadata.java

# 1.306 rxkotlin 2.2.0

## 1.306.1 Available under license :

Apache License
Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright 2012 Netflix, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.307 shared 30.3.0

## 1.307.1 Available under license :

Copyright (c) 2005-2016, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited
to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.308 kotlin-util-klib 1.5.31

## 1.308.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2018 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344952_1721124849.5868247/0/kotlin-util-klib-1-5-31-sources-jar/org/jetbrains/kotlin/library/impl/KotlinLibraryImpl.kt
No license file was found, but licenses were detected in source scan.

```
/**
* Copyright 2010-2019 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100344952_1721124849.5868247/0/kotlin-util-klib-1-5-31-sources-jar/org/jetbrains/kotlin/library/KotlinAbiVersion.kt
* /opt/cola/permits/2100344952_1721124849.5868247/0/kotlin-util-klib-1-5-31-sources-jar/org/jetbrains/kotlin/library/KotlinLibraryLayout.kt

# 1.309 jackson-databind 2.12.4

## 1.309.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and

limitations under the License.
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers.

## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0
To find the details that apply to this artifact see the accompanying LICENSE file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

# 1.310 kotlin-stdlib-jdk8 1.6.10

## 1.310.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/2100345202_1721124819.4262686/0/kotlin-stdlib-jdk8-1-6-10-sources-2-jar/kotlin/text/RegexExtensions.kt
* /opt/cola/permits/2100345202_1721124819.4262686/0/kotlin-stdlib-jdk8-1-6-10-sources-2-jar/kotlin/internal/jdk8/JDK8PlatformImplementations.kt
* /opt/cola/permits/2100345202_1721124819.4262686/0/kotlin-stdlib-jdk8-1-6-10-sources-2-jar/kotlin/collections/Collections.kt

*
/opt/cola/permits/2100345202_1721124819.4262686/0/kotlin-stdlib-jdk8-1-6-10-sources-2-jar/kotlin/streams/Streams.kt

# 1.311 ddmlib 30.3.0

## 1.311.1 Available under license :

Copyright (c) 2005-2008, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.312 kotlin-util-klib 1.7.10

## 1.312.1 Available under license :

Apache-2.0

# 1.313 kotlin-daemon-embeddable 1.7.10

## 1.313.1 Available under license :

Apache-2.0

# 1.314 kotlin-scripting-common 1.7.10

## 1.314.1 Available under license :

Apache-2.0

# 1.315 kotlin-android-extensions-runtime 1.7.10

## 1.315.1 Available under license :

Apache-2.0

# 1.316 kotlin-scripting-jvm 1.7.10

## 1.316.1 Available under license :

Apache-2.0

# 1.317 kotlin-gradle-plugin-api 1.7.10

## 1.317.1 Available under license :

Apache-2.0

# 1.318 markdown-jvm 0.2.4

## 1.318.1 Available under license :

Apache-2.0

# 1.319 android-device-provider-ddmlib-proto 30.3.0

## 1.319.1 Available under license :

Copyright (c) 2005-2021, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright
owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited
to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated

within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE
file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this
License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely
responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.
While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.320 core-common 2.1.0

## 1.320.1 Available under license :

Apache-2.0

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries.To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2024 Cisco Systems, Inc. All rights reserved.