



Configure Collection

This section contains the following topics:

- [Collection Service Overview, on page 1](#)
- [Prerequisites for Device Model Driven Telemetry, on page 1](#)
- [About Collection Jobs, on page 4](#)
- [Collection Job Payload Model, on page 5](#)
- [Create Collection Jobs, on page 10](#)
- [Best Practices and Limitations for Creating Collection Jobs, on page 10](#)
- [Collection Jobs, on page 11](#)
- [Monitoring Collection Jobs, on page 19](#)
- [List of Pre-loaded Traps and MIBs for SNMP Collection, on page 23](#)
- [List of Pre-loaded YANG Modules for MDT Collection, on page 32](#)

Collection Service Overview

Networks maintain a large amount of data that spans thousands of devices. The Cisco Crosswork Change Automation and Health Insights collects and manages that flow of data via Cisco Crosswork Data Gateway in a multi-vendor environment to provide a real-time publish/subscribe model infrastructure to Cisco Crosswork Network Automation applications. The Collection Service is highly scalable in order to meet the performance demands of the Cisco Crosswork Change Automation and Health Insights applications.

Multiple applications requesting same data overload network devices causing outages. Cisco Crosswork Data Gateway's **Collection Optimization** feature tackles this problem by optimizing collection requests. Thereby, reducing redundant data collections.

Users with administrative privileges can monitor Collection Service status and performance, and start/stop/restart it or its underlying services, using the Cisco Crosswork Change Automation and Health Insights user interface. You can also collect logs and performance metrics for this service. For help with these tasks, see [Manage Cisco Crosswork Network Automation](#).

Prerequisites for Device Model Driven Telemetry

The Cisco Crosswork Change Automation and Health Insights Collection Service configures telemetry as needed on the devices enrolled within the service.



Note If an operator configures telemetry directly on the same devices either manually or through some mechanism outside of the Collection Service, the commands must not contain the keyword `CW`. The keyword `CW` is reserved for use by the Collection Service. In particular, the following commands must not contain the keyword `CW` when configured outside of the Collection Service:

```
destination-group
sensor-group
subscription
    sensor-group-id
    destination-id
```

For example (invalid telemetry configuration):

```
telemetry model-driven
  destination-group CW_1b4ac245d863cf3e787d42bae97f1d18dd300d5e
```

For more information, see the telemetry configuration documentation for your particular device (for example: [Telemetry Configuration Guide for Cisco ASR 9000](#))

For collection to work, the maximum number of interfaces on a single device must be less than 8,000. The cumulative count of interfaces across all devices must be less than 30,000.

Invalid Telemetry Configuration

The following sample output shows an *invalid* telemetry configuration on a device when configured outside of the Collection Service.

```
telemetry model-driven
  destination-group CW_1b4ac245d863cf3e787d42bae97f1d18dd300d5e
    address-family ipv4 172.16.2.31 port 31500
      encoding self-describing-gpb
      protocol tcp
    !
  !
  destination-group CW_6da4e808ed4724911f2288dbae605bb62f9617
    address-family ipv4 <IP_address> port 31500
      encoding self-describing-gpb
      protocol tcp
    !
  !
  destination-group CW_7ee5d52d7e513640f71417d4fcd584c6a883f7c
    address-family ipv4 <IP_address> port 31500
      encoding self-describing-gpb
      protocol tcp
    !
  !
  destination-group CW_bbd6f2991d04e920fbda0f2a5ceb63d1c9f62cdf
    address-family ipv4 <IP_address> port 31500
      encoding self-describing-gpb
      protocol tcp
    !
  !
```

Valid Telemetry Configuration

The following sample output shows a *valid* telemetry configuration on a device when configured outside of the Collection Service. Note that the *<IP_address>* referred to in this sample should not be the IP address of the Cisco Crosswork Change Automation and Health Insights server. It should be the IP address of the other data consumer.

```
telemetry model-driven
destination-group CUSTOM_X_1171424d5b08d674367318299db2f8a0d7d489e9
  address family ipv4 <IP_address> port 6700
  encoding gpb
  protocol grpc no-tls
!
!
sensor-group CUSTOM_X_1171424d5b08d674367318299db2f8a0d7d489e9
  sensor-path Cisco-IOS-XR-spirit-install-instmgr-oper:software-install/active
!
!
subscription CUSTOM_X_1171424d5b08d674367318299db2f8a0d7d489e9
  sensor-group-id CUSTOM_X_1171424d5b08d674367318299db2f8a0d7d489e9 sample-interval 7000
  destination-id CUSTOM_X_1171424d5b08d674367318299db2f8a0d7d489e9
!
```



Note The **sample-interval** can be changed depending on the size of your network. It is defined in milliseconds and determines how fast you want the data to be pushed out.

Confirm that all PCCs or provider edge routers have telemetry configured and report data to . For example, routers should report prefix and tunnel counters:

```
RP/0/RP0/CPU0:PE1#show traffic-collector ipv4 counters prefix
Thu Jul 11 08:32:32.993 UTC
Prefix Label Base rate TM rate State
(Bytes/sec) (Bytes/sec)
-----
192.168.0.1/32 16001 1 0 Active
192.168.0.2/32 16002 1 0 Active
192.168.0.3/32 16003 1 0 Active
192.168.0.4/32 16004 2 0 Active
192.168.0.6/32 16006 501023 501021 Active
192.168.0.7/32 16007 17320774 17320772 Active
192.168.0.8/32 16008 3737825 3737823 Active
192.168.0.9/32 16097 3 0 Active
192.168.0.10/32 16096 2 0 Active
```

```
RP/0/RP0/CPU0:PE1#show traffic-collector ipv4 counters tunnel
Thu Jul 11 08:32:20.746 UTC
Interface Base rate Base rate State
(Packet/sec) (Bytes/sec)
-----
srte_c_102_ep_192.168.0.7 0 0 Active
```

Cisco IOS XR devices that are onboarded through telemetry must have the following configuration settings on the device to ensure that NETCONF and SSH work correctly:

```
ssh server v2
ssh server vrf default
```

```
ssh server netconf vrf default
ssh server rate-limit 600
ssh server session-limit 1024
netconf-yang agent ssh
```

Cisco IOS XR devices that are onboarded through SNMP must have SNMP enabled on the device. The following is an example of an SNMP configuration on a Cisco IOS XR device:

```
snmp-server community public RO
```

Please note that, currently, Cisco Crosswork Change Automation and Health Insights does not itself support execution of EXEC privilege commands, such as **enable**, on devices. These types of commands must be executed using the device console or other means.

About Collection Jobs

As mentioned earlier, Crosswork Data Gateway pulls functional images from the Crosswork. Each functional image represents a collection type. You can create multiple jobs for a given collection type. A collection job describes what task a Crosswork Data Gateway is expected to perform. Crosswork receives the data collection requests via these collection jobs and assigns to a Crosswork Data Gateway instance to serve the request.

You can collect more than one type of data at a time by using separate collection jobs.

For each collection job you create, Crosswork Data Gateway executes the collection request and deposits the collected data in the preferred data destination(s).

Crosswork Data Gateway lets you create three types of collection jobs:

CLI Collection Job

Enables CLI-based data collection (such as device configuration) from the network devices. The CLI collector uses XDE/PAL to collect device data for a given CLI. Only **show** commands are supported for this type of collection job.

SNMP Collection Job

Enables SNMP-based data collection based on the OIDs supported on the devices.

Supported SNMP versions include SNMPv1, SNMPv2c, and SNMPv3 for data polling and traps.

MDT Collection Job

Collects model driven telemetry data streamed from the device to the Crosswork Data Gateway.



Note

1. Crosswork Data Gateway drops incoming southbound traffic if there is no corresponding (listening) collection job request for the same. It also drops data/SNMP traps received from an unsolicited device (i.e., not attached to Crosswork Data Gateway). Crosswork Data Gateway records this in log and notifies Crosswork.
2. Polled data cannot be requested from the device until Crosswork Data Gateway is ready to process and transmit the data. If it cannot keep up with the amount of data, it sends an error to northbound interface indicating when the throttling began and condition cleared.

Collection Job Payload Model

A collection job describes the following:

- Data to be collected.
- Devices from which collection is desired and credentials to authenticate.
- Collection intervals (a periodic interval no less than 60 seconds or greater than 32 days or immediately on demand.)
- Data Destinations where the collected data is to be deposited.

A single collection job can contain either CLI commands, SNMP MIB requests, or MDT subscriptions. Crosswork Data Gateway routes the request to the appropriate collector to fetch the requested data.

A collection job has three main parts:

```
//          Device Groups(s) -- identifies the different device groups from which data is
//                               to be collected and their authentication credentials.
//          Data Destinations(s) -- identifies the different output destination the final
data                                         is sent to
//
//
```



Note Without clustered setup of external Kafka, if two destination nodes are specified, one is discarded.

Shown below is a sample collection job payload:

```
{
  "collection_job": {
    "application_context": {
      "context_id": "collection-job1",
      "application_id": "APP1"
    },
    "collection_mode": {
      "lifetime_type": "APPLICATION_MANAGED",
      "collector_type": "CLI_COLLECTOR"
    },
    "job_device_set": {
      "device_set": {
        "devices": {
          "device_ids": [
            "658adb03-cc61-448d-972f-4fcec32cbfe8"
          ]
        }
      }
    },
    "sensor_input_configs": [
      {
        "sensor_data": {
          "cli_sensor": {
            "command": "show platform"
          }
        },
        "cadence_in_millisec": "tel:60000"
      }
    ]
  }
}
```

```

    ],
    "sensor_output_configs": [
      {
        "sensor_data": {
          "cli_sensor": {
            "command": "show platform"
          }
        },
        "destination": {
          "destination_id": "1e71f2fb-ea65-4242-8efa-e33cec71b369",
          "context_id": "topic1"
        }
      }
    ]
  }
}

```

The following table explains the fields of the above payload:

Field	Type	(M)andatory or (O)ptional	Description
collection_job	json object	M	Describes collection job with application context as key.
application_context	json object	M	Unique handle to identify your application's subscription. Note Combination of application ID and Context ID must be globally unique.
application_id	String	M	Unique identifier for the application. Note The prefix "cw" is reserved for Crosswork collection jobs and must not be used for jobs of other applications.
context_id	String	M	Unique identifier for the application subscription across all the collection jobs.
collection_mode	json object	M	A json object that holds lifetime type and collector type attributes of the collection job.

Field	Type	(M)andatory or (O)ptional	Description
lifetime_type	string	O	Type of lifetime of job: <ul style="list-style-type: none"> • UNKNOWN_TEMPORAL_TYPE (default) • APPLICATION_MANAGED • CALENDAR_MANAGED • AUTO_DELETE_AFTER_N_SAMPLES
collector_type	string	M	Type of the collector: <ul style="list-style-type: none"> • UNKNOWN_COLLECTOR (default) • CLI_COLLECTOR • SNMP_COLLECTOR • TRAP_COLLECTOR • MDT_COLLECTOR
n_collections	int	M	Number of collections to run before auto-deletion. Used only when lifetime_type is AUTO_DELETE_AFTER_N_SAMPLES.
job_device_set	json object	M	A json object containing device sets information.
device_set	json object	M	Device Grouping Object. For a given request, set one of either one device grouping i.e., list of devices or a device group, but not both.
device_ids	json array	M	Array of device IDs. Corresponds to UUID of the devices in the Crosswork inventory.
sensor_input_configs	json object	M	A group of sensors and their cadences.

Field	Type	(M)andatory or (O)ptional	Description
sensor_data	json object	M	<p>Type of sensor data:</p> <ul style="list-style-type: none"> • snmp_sensor • snmp_yang_sensor • cli_yang_sensor • cli_sensor • mdt_sensor • trap_sensor • trap_yang_sensor <p>Note YANG sensors are used for Crosswork initiated collection jobs, whereas Non-YANG sensors are used for API-initiated collection jobs.</p>
cadence_in_millisecc	int	O	Optional cadence value of this sensor config in seconds. If not passed the sensor config collection is done once.
sensor_output_configs	json array	M	A group of sensors and their cadences. Sensor Output represents output of sensor to a destination for given sensor which is a topic on Kafka or gRPC server.

Field	Type	(M)andatory or (O)ptional	Description
sensor_data	json object	M	<p>Every sensor path on the input side needs to have one mapping on the output side:</p> <ul style="list-style-type: none"> • snmp_sensor • snmp_yang_sensor • cli_yang_sensor • cli_sensor • mdt_sensor • trap_sensor • trap_yang_sensor <p>Note YANG sensors are used for Crosswork initiated collection jobs, whereas Non-YANG sensors are used for API-initiated collection jobs.</p>
destination	json object	M	<p>A JSON object that holds the final data destination information. It can hold one or more type of target information. Currently supported data destinations are kafka and gRPC.</p>
destination > context_id	string	M	<p>Destination context identifier. It could be Kafka topic name if destination is external Kafka server. The combination of destination_id and context_id needs to be unique for destination.</p> <p>What context id means depends on the destination type of destination provider. If gRPC is destination type, context_id is not used and will be ignored.</p>

Field	Type	(M)andatory or (O)ptional	Description
destination_id	string	M	Unique identifier for the data destination in inventory.

Create Collection Jobs



Note Cisco Crosswork Data Gateway API access for external data collection integration is separately licensed.

You can create and delete collection jobs using the Crosswork REST APIs. To access API documentation, see [API Documentation](#).

For reference of collection job payloads, see [Collection Jobs, on page 11](#). Once the collection job is created, Crosswork adds it to the respective collector of the Crosswork Data Gateway.

The collection job is then picked up by the Crosswork Data Gateway for execution when the Image Manager syncs with Crosswork and retrieves the latest boot-config and docker-compose.

Upon successful creation of collection job, if the data destination is up, running, and accessible, Crosswork Data Gateway starts sending data to it. In this scenario, status for per device per sensor config is shown as ACTIVE in the Crosswork UI in **Collection Jobs** view. See [Monitoring Collection Jobs, on page 19](#)



Note Sensor output and input configs can be changed post creation by using a PUT collection job API. Other collection job parameters are immutable.

However, if output server is inaccessible, Crosswork Data Gateway fails to send data to it.

If you want to delete a collection job created by a Cisco Crosswork Change Automation and Health Insights application, it must be deleted via the corresponding application only.

Upon job deletion, the collector sends an alert to the Crosswork for that job once all the devices per sensor path execution has stopped that can be viewed in the Crosswork UI at **Admin > Collection Jobs**.

Best Practices and Limitations for Creating Collection Jobs

Cisco recommends that following best practices be followed while creating collection job payloads:

Limitation	Best Practices
Scale	

Limitation	Best Practices
<p>Maximum size of sensor path collected data is 10 MB. Crosswork Data Gateway collector data at the same time from N devices cannot exceed 6 GB.</p>	<ol style="list-style-type: none"> Any collected sensor path's data size should not be > 10 MB when data destination is Kafka, otherwise you will get the following error message in <code>collector.log</code> file (to access log files, see Run show-tech): RecordTooLargeException: The message is xxxxxxxx bytes when serialized which is larger than the maximum request size you have configured with the max.request.size configuration. Crosswork Data Gateway collector data at the same time from N devices with M sensor path per device and with P average size of collected data per sensor path cannot exceed 6 GB i.e., (N = # of devices) x (M = # of sensor path) x (P= Average Message size per sensor path) < 6GB
Log Purge Policy	
<p>When total log file size for a collector reaches 2 GB, Crosswork Data Gateway starts cleaning up by removing the old log files.</p>	<p>If you would like to save the old log files, save it to any other remote server before the total log file size reaches 2 GB per collector. This can be done by running show-tech from Crosswork Data Gateway Main Menu > Troubleshooting.</p>

Collection Jobs

This section contains sample collection job payloads for the following collection profiles:

- [CLI Collection Job, on page 11](#)
- [SNMP Collection Jobs, on page 13](#)
- [MDT Collection Job, on page 17](#)

CLI Collection Job

Crosswork Data Gateway supports CLI-based data collection from the network devices. It uses XDE/PAL to collect device data for a given CLI. Only show commands are supported for this type of collection job.

**Note**

- While creating a CLI collection job, you must validate it for any errors. Crosswork Data Gateway does not validate collection jobs. Hence, even though a collection job is invalid, its status is still shown as **Active** on **Collection Jobs** page in the Crosswork UI.
- Device should not have any banner configuration for CLI collection to work properly. Please refer to device documentation on how to turn this off.
- The value of **Cadence** is in seconds. It should be set either to 0 to indicate the sensor configured to be collected only once.

OR

It should be ≥ 60 (i.e. at least 1 minute) up to 2764800 seconds (i.e. at most 32 days) max, indicating how frequently configured sensor data should be collected.

Following is a CLI collection job sample:

```
{
  "collection_job": {
    "application_context": {
      "context_id": "collection-job1",
      "application_id": "APP1"
    },
    "collection_mode": {
      "lifetime_type": "APPLICATION_MANAGED",
      "collector_type": "CLI_COLLECTOR"
    },
    "job_device_set": {
      "device_set": {
        "devices": {
          "device_ids": [
            "658adb03-cc61-448d-972f-4fcec32cbfe8"
          ]
        }
      }
    },
    "sensor_input_configs": [
      {
        "sensor_data": {
          "cli_sensor": {
            "command": "show platform"
          }
        },
        "cadence_in_millisec": "tel:60000"
      }
    ],
    "sensor_output_configs": [
      {
        "sensor_data": {
          "cli_sensor": {
            "command": "show platform"
          }
        },
        "destination": {
          "destination_id": "1e71f2fb-ea65-4242-8efa-e33cec71b369",
          "context_id": "topic1"
        }
      }
    ]
  }
}
```

```
}
}
```

SNMP Collection Jobs

Crosswork Data Gateway supports SNMP-based data collection based on the OIDs supported on the devices.

The SNMP collector makes a poll request to Crosswork to get its configuration profile (a list of MIB objects to collect and a list of devices to fetch from). It determines the corresponding OIDs by looking up the pre-packaged list of MIB modules or the custom list of MIB modules.



Note MIBs are required only if the collection request references MIB TABLE names or SCALAR names. However, if the requests are OID-based, then MIBs are not required.

Once the OIDs are resolved, they are provided as input to the SNMP collectors.

The device packages can be imported into the Crosswork Data Gateway VM as described in Section [Add a Custom Software Package](#).

The following SNMP versions are supported:

- SNMPv1
- SNMPv2c
- SNMPv3

The table below lists supported privacy protocols and the value that needs to be given in the collection payload for SNMP and SNMP Trap collection jobs:

Protocol	SNMP Collection Payload	SNMP Trap Collection Payload
aes	AES	N/A
des56	DES	DES
3des	3DES	3DES
aes 128	AES128	AES128
aes 192	AES192 or CiscoAES192(Cisco specific)	AES192 or CiscoAES192(Cisco specific)
aes 256	AES256 or CiscoAES256(Cisco specific)	AES256 or CiscoAES256(Cisco specific)

**Note**

- While creating an SNMP collection job, you must validate it for any errors. Crosswork Data Gateway does not validate collection jobs. Hence, even though a collection job is invalid, its status is still shown as **Active**.
- The value of **Cadence** is in seconds. It should be set either to 0 to indicate the sensor configured to be collected only once.

OR

It should be ≥ 60 (i.e. at least 1 minute) up to 2764800 seconds (i.e. at most 32 days) max, indicating how frequently configured sensor data should be collected.

- When collection from a device is skipped due to previous execution still in progress, Crosswork Data Gateway raises a warning log. No alert is generated for this scenario.
- For SNMP v1/v2c, if the device details (such as host or community string) are incorrect in the payload, Crosswork Data Gateway ignores the traps received from the device and logs the a WARN message.

In case of SNMP v3, if the device details (such as auth, priv, and security name details) are incorrect in the payload, Crosswork Data Gateway filters it out and hence, does not receive the trap. Thus, no WARN message is logged.

Sample Configurations on Device:

Version	Configuration
V1	<pre>snmp-server group group1 v1 snmp-server user user1 group1 v1 snmp-server host <host_ip> traps <community_string> udp-port 1062</pre> <p>For example,</p> <pre>snmp-server host 172.29.194.78 traps test udp-port 1062</pre> <p>Note Version 1 is the default version used by the device.</p>
V2c	<pre>snmp-server group group1 v2c snmp-server user user1 group1 v2c snmp-server host 172.29.194.142 traps version 2c v2test udp-port 1062</pre>
V3	<pre>snmp-server group group1 v3 auth notify user1 read user1 write user1 snmp-server view user1 1.3 included snmp-server user user1 group1 v3 auth md5 <password> priv aes 128 <password> snmp-server host 172.23.92.193 traps version 3 priv user1 udp-port 1062</pre>

The SNMP Collector supports the following operations:

- SCALAR

- MIB_WALK
- TRAP
- DEVICE_PACKAGE

These operations are defined in the sensor config (see payload sample below).



Note There is an optional **deviceParams** attribute **snmpRequestTimeoutMillis** (not shown in the sample payloads) that should be used if the device response time is very high. It's not recommended to use **snmpRequestTimeoutMillis** unless you are absolutely certain that your device response time is very high.

The value for **snmpRequestTimeoutMillis** should be specified in milliseconds:

Default value is 1500 milliseconds

Minimum value is 1500 milliseconds

However, there is no limitation on the maximum value of this attribute.

Following is an SNMP collection job sample:

```
{
  "collection_job": {
    "application_context": {
      "context_id": "collection-job1",
      "application_id": "APP1"
    },
    "collection_mode": {
      "lifetime_type": "APPLICATION_MANAGED",
      "collector_type": "SNMP_COLLECTOR"
    },
    "job_device_set": {
      "device_set": {
        "devices": {
          "device_ids": [
            "c70fc034-0cbd-443f-ad3d-a30d4319f937",
            "8627c130-9127-4ed7-ace5-93d3b4321d5e",
            "c0067069-c8f6-4183-9e67-1f2e9bf56f58"
          ]
        }
      }
    },
    "sensor_input_configs": [
      {
        "sensor_data": {
          "snmp_sensor": {
            "snmp_mib": {
              "oid": "1.3.6.1.2.1.1.3.0",
              "snmp_operation": "SCALAR"
            }
          }
        },
        "cadence_in_millisecc": "60000"
      },
      {
        "sensor_data": {
          "snmp_sensor": {
            "snmp_mib": {
              "oid": "1.3.6.1.2.1.31.1.1",
              "snmp_operation": "TABLE"
            }
          }
        }
      }
    ]
  }
}
```

```

    }
  },
  "cadence_in_millisec": "60000"
}
],
"sensor_output_configs": [
  {
    "sensor_data": {
      "snmp_sensor": {
        "snmp_mib": {
          "oid": "1.3.6.1.2.1.1.3.0",
          "snmp_operation": "SCALAR"
        }
      }
    },
    "destination": {
      "destination_id": "4c2ab662-2670-4b3c-b7d3-b94acba98c56",
      "context_id": "topic1_461cb8aa-a16a-44b8-b79f-c3daf3ea925f"
    }
  },
  {
    "sensor_data": {
      "snmp_sensor": {
        "snmp_mib": {
          "oid": "1.3.6.1.2.1.31.1.1",
          "snmp_operation": "TABLE"
        }
      }
    },
    "destination": {
      "destination_id": "4c2ab662-2670-4b3c-b7d3-b94acba98c56",
      "context_id": "topic2_e7ed6300-fc8c-47ee-8445-70e543057f8a"
    }
  }
]
}
}

```

SNMP Traps Collection Job

SNMP traps are handled in a similar manner. Trap listeners listen on a port and then dispatch data to recipients (based on their topic of interest).



Note

- Device should have been pre-configured by the traps.
- Crosswork Data Gateway listens on UDP port 1062 for Traps.
- Collection jobn gets stuck in Accepted state when wrong authentication credentials are passed for SNMP Traps.
- For list of supported Traps and MIBs, see [List of Pre-loaded Traps and MIBs for SNMP Collection, on page 23](#).

On receiving a trap, Crosswork Data Gateway does the following validations:

1. Check if any collection job is created for the device.
2. Checks the trap version and community string.

- For SNMP v3, validates for user auth and priv protocol and credentials.

Following is an SNMP-Trap collection job sample:

```
{
  "collection_job": {
    "application_context": {
      "context_id": "collection-job1",
      "application_id": "APP1"
    },
    "collection_mode": {
      "lifetime_type": "APPLICATION_MANAGED",
      "collector_type": "TRAP_COLLECTOR"
    },
    "job_device_set": {
      "device_set": {
        "devices": {
          "device_ids": [
            "a9b8f43d-130b-4866-a26a-4d0f9e07562a",
            "8c4431a0-f21d-452d-95a8-84323a19e0d6",
            "eaab2647-2351-40ae-bf94-6e4a3d79af3a"
          ]
        }
      }
    },
    "sensor_input_configs": [
      {
        "sensor_data": {
          "trap_sensor": {
            "path": "1.3.6.1.6.3.1.1.4"
          }
        },
        "cadence_in_millisecc": "60000"
      }
    ],
    "sensor_output_configs": [
      {
        "sensor_data": {
          "trap_sensor": {
            "path": "1.3.6.1.6.3.1.1.4"
          }
        },
        "destination": {
          "destination_id": "4c2ab662-2670-4b3c-b7d3-b94acba98c56",
          "context_id": "topic1_696600ae-80ee-4a02-96cb-3a01a2415324"
        }
      }
    ]
  }
}
```

MDT Collection Job

Crosswork Data Gateway supports data collection from network devices using Model-driven Telemetry (MDT) to consume telemetry streams directly from devices.

**Note**

- MDT collector retains the collection ID that comes as part of the telemetry proto for the device. This behavior is different from CLI and SNMP collectors which compute the collection ID based on the sequence number of the collection.
- MDT collection jobs require some configuration to be done on the device. This configuration is automatically taken care of by NSO.
- If there is some change (delete/update) in existing MDT jobs between backup and restore operations, Crosswork does not replay the jobs for config update on the devices as it involves Provider(NSO). You have to restore configs on provider/devices. Crosswork will just restore the jobs in database.
- Before using any YANG modules, check if they are supported. See Section: [List of Pre-loaded YANG Modules for MDT Collection](#) , on page 32.

It supports data collection for the following transport mode:

- MDT TCP Dial-out Mode

Following is a sample of MDT collection payload:

```
{
  "collection_job": {
    "job_device_set": {
      "device_set": {
        "device_group": "mdt"
      }
    },
    "sensor_output_configs": [{
      "sensor_data": {
        "mdt_sensor": {
          "path":
"Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters"

        }
      },
      "destination": {
        "context_id": "cw.mdt_sensor.cisco-ios-xr-infra-statsd-oper.gpb",
        "destination_id": "c2a8fba8-8363-3d22-b0c2-a9e449693fae"
      }
    }],
    "sensor_data": {
      "mdt_sensor": {
        "path": "Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/data-rate"
      }
    },
    "destination": {
      "context_id": "cw.mdt_sensor.cisco-ios-xr-infra-statsd-oper.gpb",
      "destination_id": "c2a8fba8-8363-3d22-b0c2-a9e449693fae"
    }
  }],
  "sensor_input_configs": [{
    "sensor_data": {
      "mdt_sensor": {
        "path": "Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/data-rate"
```

```

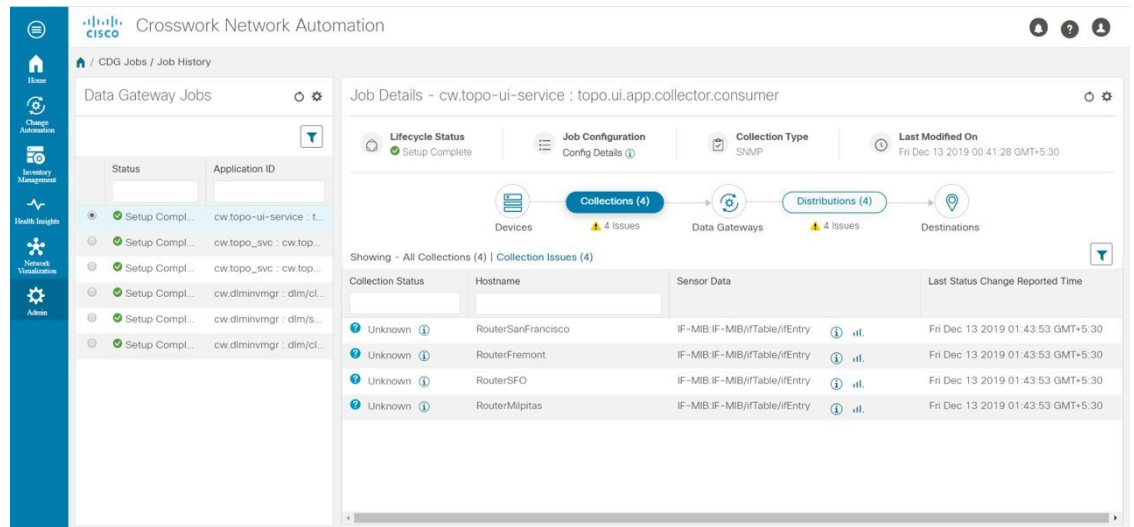
    },
    "cadence_in_millise": "70000"
  }, {
    "sensor_data": {
      "mdt_sensor": {
        "path":
"Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters"
      }
    }
  },
  "cadence_in_millise": "70000"
}
],
"application_context": {
  "context_id": "c4",
  "application_id": "a4-mdt"
},
"collection_mode": {
  "lifetime_type": "APPLICATION_MANAGED",
  "collector_type": "MDT_COLLECTOR"
}
}
}
}

```

Monitoring Collection Jobs

From the **Collection Jobs** view, you can monitor the status of the collection jobs currently active on all the Cisco Crosswork Data Gateway instances enrolled with Cisco Crosswork Change Automation and Health Insights, such as system jobs and API-defined collection jobs.

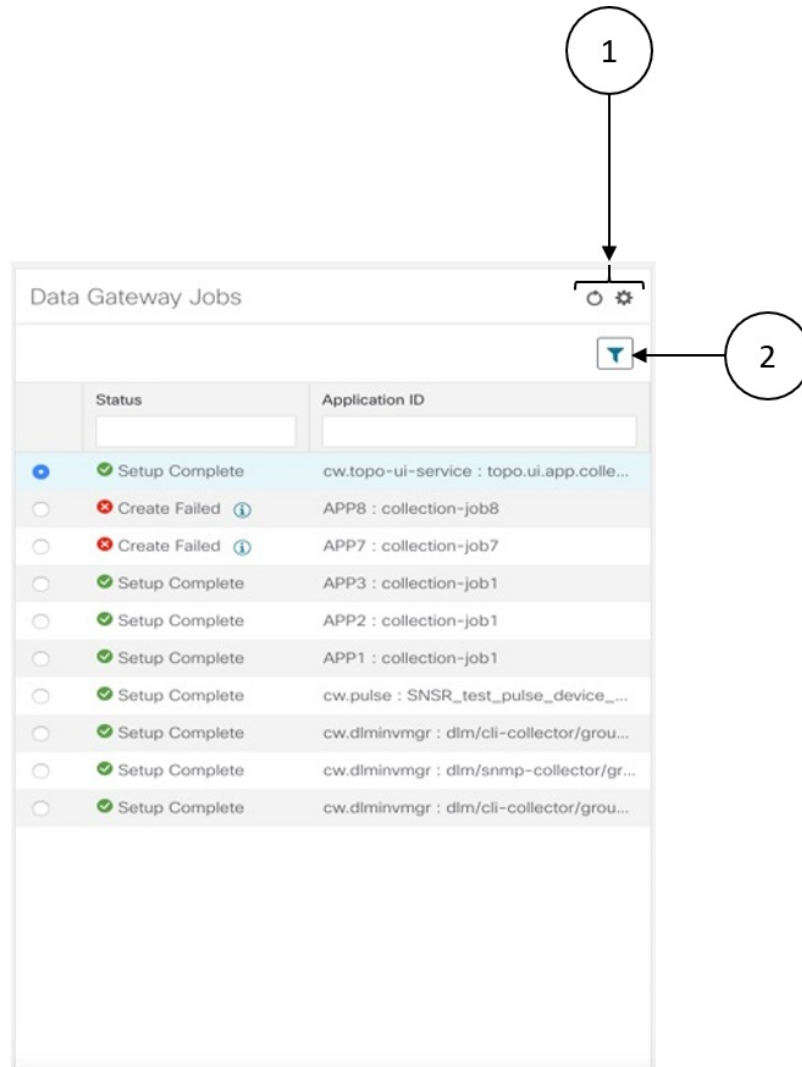
From the navigation bar, choose **Admin > Collection Jobs**.






Item	Description
Data Gateway Jobs Pane	Shows the list of all active collection jobs along with their status and application ID.

Item	Description
Job Details Pane	Shows the details of a particular job selected in the Data Gateway Jobs pane.

To view details of a collection job, select the collection job from the **Data Gateway Jobs** pane. The details of the selected job are displayed in the **Job Details** pane right next to the **Data Gateway Jobs** pane.



Item	Description
1	Click  to refresh the Data Gateway Jobs window.
	Click  to choose the columns to make visible in the Data Gateway Jobs window (see Set, Sort and Filter Table Data).

Item	Description
2	Click  to set filter criteria on one or more columns in the Data Destinations window.
	Click the Clear All Filters link to clear any filter criteria you may have set.

Data Gateway Jobs pane displays only the status and application ID.




Note

- Create Failed error means out of N devices , some devices detup failed. However, the collection would happen on the other devices. You can identify the device(s) causing this error by using Control Status API.
- If job creation failed on a particular device because of NSO errors, after fixing NSO errors , you have to manually change the administration state of the device first to "Down" and then "Up". However, doing so resets the collection on the device.



Note

Create/Delete failed errors are shown in a different screen pop up. Click  to the job status to see details of the error.

- You may also try recreating the job using PUT collection job API with the same payload.





However, when you select a job, more details are displayed in the **Job Details** pane:

The screenshot shows the 'Job Details' pane for a job named 'cw.topo-ui-service : topo.ui.app.collector.consumer'. The pane includes several sections:


- 1**: Job Details header.
- 2**: Lifecycle Status (Setup Complete).
- 3**: Job Configuration (Config Details).
- 4**: Collection Type (SNMP).
- 5**: Last Modified On (Thu Dec 12 2019 11:11:28 PST).
- 8**: Settings icon.
- 6**: Collections (4) with 4 Issues.
- 7**: Distributions (4) with 4 Issues.
- 9**: Filter icon in the table header.



The table below shows the data for Collections:

Collection Status	Hostname	Device Id	Sensor Data	Last Status Change Reported Time
Unknown	RouterSanFrancisco	d112738f-c...	IF-MIB:IF-MIB:ifTable/IfE...	Thu Dec 12 2019 12:13:53 PST
Unknown	RouterFremont	c700dde9-...	IF-MIB:IF-MIB:ifTable/IfE...	Thu Dec 12 2019 12:13:53 PST
Unknown	RouterSFO	999d81a9-...	IF-MIB:IF-MIB:ifTable/IfE...	Thu Dec 12 2019 12:13:53 PST
Unknown	RouterMilpitas	618d7e4f-2...	IF-MIB:IF-MIB:ifTable/IfE...	Thu Dec 12 2019 12:13:53 PST

Item	Description
1	Application name and context associated with the collection job.
2	Lifecycle status of the collection job.
3	Job payload of the collection job that you pass in the REST API request. Click  icon next to Config Details to view the job configuration. Crosswork Data Gateway lets you view configuration in two modes: <ul style="list-style-type: none"> • View Mode • Text Mode
4	Collection Type
5	Time and date of last modification of the collection job.
6	Collections (x): x refers to requested input collections that span device by sensor paths. The corresponding (y) Issues is the count of input collections that are in UNKNOWN or FAILED state.
7	Distributions (x): x refers to requested output collections that span device by sensor paths. The corresponding (y) Issues is the count of output collections that are in UNKNOWN or FAILED state.
8	Click  to refresh the Job Details window.
	Click  to choose the columns to make visible in the Job Details window (see Set, Sort and Filter Table Data).
9	Click  to set filter criteria on one or more columns in the Job Details window.
	Click the Clear All Filters link to clear any filter criteria you may have set.

Job Details pane displays the following details about a collection job:

Field	Description
Collection/Distribution Status	Status of the collection/distribution. It is reported on a on change basis from Cisco Crosswork Data Gateway. Click  next to the collection/distribution status for details.
Hostname	Application with which the collection job is associated.
Device Id	Unique identifier of the device from which data is being collected.

Field	Description
Sensor Data	<p>Sensor path</p> <p>Click  to see collection/distribution summary. From the sensor data summary pop up you can copy the sensor data by clicking Copy to Clipboard.</p> <p>and</p> <p>Click  to see collection/distribution metrics summary. The metrics are reported on cadence-basis i.e., once every 10 minutes by default. It shows the following metrics for a collection:</p> <ul style="list-style-type: none"> • last_collection_time_msec • total_collection_message_count • last_device_latency_msec • last_collection_cadence_msec <p>It shows the following metrics for a collection:</p> <ul style="list-style-type: none"> • total_output_message_count • last_destination_latency_msec • last_output_cadence_msec • last_output_time_msec • total_output_bytes_count
Destination	Data destination for the job.
Last Status Change Reported Time	Time and date on which last status change was reported for that device sensor pair from Cisco Crosswork Data Gateway.

List of Pre-loaded Traps and MIBs for SNMP Collection

This section lists the Traps and MIBs that the Collection Service supports for SNMP collection.



Note This list is applicable only when Crosswork is the target application and is not limited when the target is an external application.

Note the following constraints:

- The system cannot extract index values from OIDs of conceptual tables. If any of the columns that define indices in the conceptual table are not populated, the index value is replaced on the data plane with the instance identifier (oid suffix) of the row.
- The system cannot extract index values from conceptual tables that include the **AUGMENT** keyword or refer to indices of other tables.
- Named-number enumerations (using the integer syntax) are sent on the wire using their numeric value.

Table 1: Supported Yangified Traps

Trap	OID
linkDown	1.3.6.1.6.3.1.1.5.3
linkUp	1.3.6.1.6.3.1.1.5.4
coldStart	1.3.6.1.6.3.1.1.5.1
isisAdjacencyChange	1.3.6.1.2.1.138.0.17

ADSL-LINE-MIB.mib	CISCO-UBE-MIB.mib	JUNIPER-ANALYZER-MIB.mib
ADSL-TC-MIB.mib	CISCO-UNIFIED-COMPUTING-ADAPTOR-MIB.mib	JUNIPER-ATM-COS-MIB.mib
AGENTX-MIB.mib	CISCO-UNIFIED-COMPUTING-COMPUTE-MIB.mib	JUNIPER-ATM-MIB.mib
AIRSPACE-REF -MIB.mib	CISCO-UNIFIED-COMPUTING-ETHER-MIB.mib	JUNIPER-BFD-MIB.mib
AIRSPACE-SWITCHING -MIB.mib	CISCO-UNIFIED-COMPUTING-FC-MIB.mib	JUNIPER-CFGMGMT -MIB.mib
AIRSPACE-WIRELESS -MIB.mib	CISCO-UNIFIED-COMPUTING-MEMORY-MIB.mib	JUNIPER-CHASSIS-DEFINES -MIB.mib
ALARM-MIB.mib	CISCO-UNIFIED-COMPUTING-MIB.mib	JUNIPER-CHASSIS-FWDD -MIB.mib
ALCATEL-IEEE8021 -PAE-MIB.mib	CISCO-UNIFIED-COMPUTING-NETWORK-MIB.mib	JUNIPER-COLLECTOR-MIB.mib
APS-MIB.mib	CISCO-UNIFIED-COMPUTING-PROCESSOR-MIB.mib	JUNIPER-COS-MIB.mib
ARUBA-TC.mib	CISCO-UNIFIED-COMPUTING-TC-MIB.mib	JUNIPER-DCU-MIB.mib
ATM-FORUM-MIB.mib	CISCO-VLAN-IFTABLE-RELATIONSHIP-MIB.mib	JUNIPER-DFC-MIB.mib
ATM-FORUM-TC-MIB.mib	CISCO-VLAN-MEMBERSHIP-MIB.mib	JUNIPER-DOM-MIB.mib
ATM-MIB.mib	CISCO-VOICE-COMMON-DIAL-CONTROL-MIB.mib	JUNIPER-EVENT-MIB.mib

ATM-TC-MIB.mib	CISCO-VOICE-DIAL-CONTROL-MIB.mib	JUNIPER-EX-MAC-NOTIFICATION-MIB.mib
ATM2-MIB.mib	CISCO-VOICE-DNIS-MIB.mib	JUNIPER-EX-SMI.mib
BFD-STD-MIB.mib	CISCO-VPDN-MGMT- MIB.mib	JUNIPER-EXPERIMENT-MIB.mib
BGP4-MIB.mib	CISCO-VTP-MIB.mib	JUNIPER-FIREWALL- MIB.mib
BGP4-V2-MIB-JUNIPER.mib	CISCO-WIRELESS-NOTIFICATION-MIB.mib	JUNIPER-HOSTRESOURCES-MIB.mib
BRIDGE-MIB.mib	CISCOSB-DEVICEPARAMS-MIB.mib	JUNIPER-IF-MIB.mib
CISCO-AAA-SERVER -MIB.mib	CISCOSB-HWENVIRONMENT.mib	JUNIPER-IPFORWARD-MIB.mib
CISCO-AAA-SESSION -MIB.mib	CISCOSB-MIB.mib	JUNIPER-IPSEC-FLOW-MON-MIB.mib
CISCO-AAL5-MIB.mib	CISCOSB-Physicaldescription-MIB.mib	JUNIPER-IPv4-MIB.mib
CISCO-ACCESS-ENVMON-MIB.mib	COGNIO-SMI.mib	JUNIPER-IPv6-MIB.mib
CISCO-ATM-EXT-MIB.mib	COGNIO-TRAPS-MIB.mib	JUNIPER-JDHCP-MIB.mib
CISCO-ATM-PVCTRAP-EXTN-MIB.mib	DIAL-CONTROL-MIB.mib	JUNIPER-JDHCPV6-MIB.mib
CISCO-ATM-QOS-MIB.mib	DIFFSERV-DSCP-TC.mib	JUNIPER-JS-AUTH-MIB.mib
CISCO-AUTH-FRAMEWORK-MIB.mib	DIFFSERV-MIB.mib	JUNIPER-JS-CERT-MIB.mib
CISCO-BGP-POLICY-ACCOUNTING-MIB.mib	DISMAN-EVENT-MIB.mib	JUNIPER-JS-DNS-MIB.mib
CISCO-BGP4-MIB.mib	DISMAN-EXPRESSION-MIB.mib	JUNIPER-JS-IDP-MIB.mib
CISCO-BULK-FILE-MIB.mib	DISMAN-NSLOOKUP-MIB.mib	JUNIPER-JS-IF-EXT-MIB.mib
CISCO-CBP-TARGET-MIB.mib	DISMAN-PING-MIB.mib	JUNIPER-JS-IPSEC-VPN-MIB.mib
CISCO-CBP-TARGET-TC-MIB.mib	DISMAN-SCHEDULE-MIB.mib	JUNIPER-JS-NAT-MIB.mib
CISCO-CBP-TC-MIB.mib	DISMAN-SCRIPT-MIB.mib	JUNIPER-JS-POLICY-MIB.mib
CISCO-CCME-MIB.mib	DISMAN-TRACEROUTE-MIB.mib	JUNIPER-JS-SCREENING-MIB.mib
CISCO-CDP-MIB.mib	DOT3-OAM-MIB.mib	JUNIPER-JS-SMI.mib
CISCO-CEF-MIB.mib	DRAFT-MSDP-MIB.mib	JUNIPER-L2ALD-MIB.mib
CISCO-CEF-TC.mib	DS0-MIB.mib	JUNIPER-L2CP-FEATURES-MIB.mib

CISCO-CLASS-BASED-QOS-MIB.mib	DS1-MIB.mib	JUNIPER-LDP-MIB.mib
CISCO-CONFIG-COPY -MIB.mib	DS3-MIB.mib	JUNIPER-MAC-MIB.mib
CISCO-CONFIG-MAN -MIB.mib	ENTITY-MIB.mib	JUNIPER-MAG-MIB.mib
CISCO-CONTENT-ENGINE-MIB.mib	ENTITY-SENSOR-MIB.mib	JUNIPER-MIB.mib
CISCO-CONTEXT-MAPPING-MIB.mib	ENTITY-STATE-MIB.mib	JUNIPER-MIMSTP-MIB.mib
CISCO-DATA-COLLECTION-MIB.mib	ENTITY-STATE-TC-MIB.mib	JUNIPER-MPLS-LDP-MIB.mib
CISCO-DEVICE-EXCEPTION-REPORTING-MIB.mib	ESO-CONSORTIUM-MIB.mib	JUNIPER-OTN-MIB.mib
CISCO-DIAL-CONTROL-MIB.mib	ETHER-WIS.mib	JUNIPER-PAE-EXTENSION-MIB.mib
CISCO-DOT11-ASSOCIATION-MIB.mib	EXPRESSION-MIB.mib	JUNIPER-PFE-MIB.mib
CISCO-DOT11-HT-PHY-MIB.mib	EtherLike-MIB.mib	JUNIPER-PING-MIB.mib
CISCO-DOT11-IF-MIB.mib	FDDI-SMT73-MIB.mib	JUNIPER-PMon-MIB.mib
CISCO-DOT11-SSID-SECURITY-MIB.mib	FR-MFR-MIB.mib	JUNIPER-POWER-SUPPLY-UNIT-MIB.mib
CISCO-DOT3-OAM-MIB.mib	FRAME-RELAY-DTE-MIB.mib	JUNIPER-PW-ATM-MIB.mib
CISCO-DS3-MIB.mib	FRNETSERV-MIB.mib	JUNIPER-PW-TDM-MIB.mib
CISCO-DYNAMIC-TEMPLATE-MIB.mib	GMPLS-LSR-STD-MIB.mib	JUNIPER-RMON-MIB.mib
CISCO-DYNAMIC-TEMPLATE-TC-MIB.mib	GMPLS-TC-STD-MIB.mib	JUNIPER-RPF-MIB.mib
CISCO-EIGRP-MIB.mib	GMPLS-TE-STD-MIB.mib	JUNIPER-RPM-MIB.mib
CISCO-EMBEDDED-EVENT-MGR-MIB.mib	HC-PerfHist-TC-MIB.mib	JUNIPER-RPS-MIB.mib
CISCO-ENHANCED-IMAGE-MIB.mib	HC-RMON-MIB.mib	JUNIPER-RSVP-MIB.mib
CISCO-ENHANCED-MEMPOOL-MIB.mib	HCNUM-TC.mib	JUNIPER-SCU-MIB.mib
CISCO-ENTITY-ASSET-MIB.mib	HOST-RESOURCES-MIB.mib	JUNIPER-SECURE-ACCESSPORT-MIB.mib
CISCO-ENTITY-EXT-MIB.mib	HOST-RESOURCES-TYPES.mib	JUNIPER-SMI.mib
CISCO-ENTITY-FRU-CONTROL-MIB.mib	HUAWEI-ALARM-RELIABILITY-MIB.mib	JUNIPER-SONET-MIB.mib
CISCO-ENTITY-QFP-MIB.mib	HUAWEI-BASE-TRAP-MIB.mib	JUNIPER-SP-MIB.mib
CISCO-ENTITY-REDUNDANCY-MIB.mib	HUAWEI-BFD-MIB.mib	JUNIPER-SRX3000-SUP-MONITORING-MIB.mib

CISCO-ENTITY-REDUNDANCY-TC-MIB.mib	HUAWEI-BGP-ACCOUNTING-MIB.mib	JUNIPER-SYSLOG-MIB.mib
CISCO-ENTITY-SENSOR-MIB.mib	HUAWEI-BGP-GR-MIB.mib	JUNIPER-TRACEROUTE-MIB.mib
CISCO-ENTITY-VENDOR-TYPE-OID-MIB.mib	HUAWEI-BGP-VPN-MIB.mib	JUNIPER-USER-AAA-MIB.mib
CISCO-ENVMON-MIB.mib	HUAWEI-BRAS-L2TP-MIB.mib	JUNIPER-UTIL-MIB.mib
CISCO-EPM-NOTIFICATION-MIB.mib	HUAWEI-BRAS-MULTICAST-MIB.mib	JUNIPER-VIRTUALCHASSIS-MIB.mib
CISCO-ETHER-CFM-MIB.mib	HUAWEI-BRAS-PPPoX-MIB.mib	JUNIPER-VLAN-MIB.mib
CISCO-ETHERLIKE-EXT-MIB.mib	HUAWEI-BRAS-RFC3548-MIB.mib	JUNIPER-VPN-MIB.mib
CISCO-FABRIC-C12K-MIB.mib	HUAWEI-BRAS-USERVLAN-MIB.mib	L2TP-MIB.mib
CISCO-FIREWALL-TC.mib	HUAWEI-BULKSTAT-MIB.mib	LANGTAG-TC-MIB.mib
CISCO-FLASH-MIB.mib	HUAWEI-CCC-MIB.mib	LLDP-EXT-DOT1-MIB.mib
CISCO-FRAME-RELAY-MIB.mib	HUAWEI-CE-PING-MIB.mib	LLDP-EXT-DOT3-MIB.mib
CISCO-FTP-CLIENT-MIB.mib	HUAWEI-CLOCK-MIB.mib	LLDP-MIB.mib
CISCO-HSRP-EXT-MIB.mib	HUAWEI-CONFIG-MAN-MIB.mib	MAU-MIB.mib
CISCO-HSRP-MIB.mib	HUAWEI-CPU-MIB.mib	MGMD-STD-MIB.mib
CISCO-IETF-ATM2-PVCTRAP-MIB.mib	HUAWEI-DATASYNC-MIB.mib	MPLS-FTN-STD-MIB.mib
CISCO-IETF-BFD-MIB.mib	HUAWEI-DC-TRAP-MIB.mib	MPLS-L3VPN-STD-MIB.mib
CISCO-IETF-FRR-MIB.mib	HUAWEI-DEVICE-EXT-MIB.mib	MPLS-LDP-ATM-STD-MIB.mib
CISCO-IETF-IPMROUTE-MIB.mib	HUAWEI-DEVICE-MIB.mib	MPLS-LDP-FRAME-RELAY-SID-MIB.mib
CISCO-IETF-ISIS-MIB.mib	HUAWEI-DHCPR-MIB.mib	MPLS-LDP-GENERIC-STD-MIB.mib
CISCO-IETF-MPLS-ID-STD-03-MIB.mib	HUAWEI-DIALER-MIB.mib	MPLS-LDP-MIB.mib
CISCO-IETF-MPLS-TE-EXT-SID-03-MIB.mib	HUAWEI-DLDP-MIB.mib	MPLS-LDP-STD-MIB.mib
CISCO-IETF-MPLS-TE-P2MP-SID-MIB.mib	HUAWEI-E-TRUNK-MIB.mib	MPLS-LSR-MIB.mib
CISCO-IETF-MSDP-MIB.mib	HUAWEI-ENTITY-TRAP-MIB.mib	MPLS-LSR-STD-MIB.mib
CISCO-IETF-PIM-EXT-MIB.mib	HUAWEI-EPON-MIB.mib	MPLS-MIB.mib
CISCO-IETF-PIM-MIB.mib	HUAWEI-FIREWALL-MIB.mib	MPLS-TC-MIB.mib
CISCO-IETF-PW-ATM-MIB.mib	HUAWEI-FR-QOS-MIB.mib	MPLS-TC-STD-MIB.mib
CISCO-IETF-PW-ENET-MIB.mib	HUAWEI-FTP-MIB.mib	MPLS-TE-MIB.mib
CISCO-IETF-PW-MIB.mib	HUAWEI-GTL-MIB.mib	MPLS-TE-STD-MIB.mib
CISCO-IETF-PW-MPLS-MIB.mib	HUAWEI-GTSM-MIB.mib	MPLS-VPN-MIB.mib
CISCO-IETF-PW-TC-MIB.mib	HUAWEI-HGMP-MIB.mib	MSDP-MIB.mib
CISCO-IETF-PW-TDM-MIB.mib	HUAWEI-HTTP-MIB.mib	NET-SNMP-AGENT-MIB.mib
CISCO-IETF-VPLS-BGP-EXT-MIB.mib	HUAWEI-HWTACACS-MIB.mib	NET-SNMP-EXAMPLES-MIB.mib

CISCO-IETF-VPLS-GENERIC-MIB.mib	HUAWEI-IF-QOS-MIB.mib	NET-SNMP-MIB.mib
CISCO-IETF-VPLS-LDP-MIB.mib	HUAWEI-IMA-MIB.mib	NET-SNMP-TC.mib
CISCO-IF-EXTENSION-MIB.mib	HUAWEI-IPV6-MIB.mib	NHRP-MIB.mib
CISCO-IGMP-FILTER-MIB.mib	HUAWEI-KOMPELLA-MIB.mib	NOTIFICATION-LOG-MIB.mib
CISCO-IMAGE-LICENSE-MGMT-MIB.mib	HUAWEI-L2IF-MIB.mib	OLD-CISCO-CHASSIS-MIB.mib
CISCO-IMAGE-MIB.mib	HUAWEI-L2MULTICAST-MIB.mib	OLD-CISCO-INTERFACES-MIB.mib
CISCO-IMAGE-TC.mib	HUAWEI-L2TP-EXT-MIB.mib	OLD-CISCO-SYS-MIB.mib
CISCO-IP-LOCAL-POOL-MIB.mib	HUAWEI-L2VPN-MIB.mib	OLD-CISCO-SYSTEM-MIB.mib
CISCO-IP-TAP-MIB.mib	HUAWEI-L3VLAN-MIB.mib	OPT-IF-MIB.mib
CISCO-IP-URPF-MIB.mib	HUAWEI-LDT-MIB.mib	ORINOCO-MIB.mib
CISCO-IPMROUTE-MIB.mib	HUAWEI-LINE-MIB.mib	OSPF-MIB.mib
CISCO-IPSEC-FLOW-MONITOR-MIB.mib	HUAWEI-LLDP-MIB.mib	OSPF-TRAP-MIB.mib
CISCO-IPSEC-MIB.mib	HUAWEI-LOOPDETECT-MIB.mib	OSPFV3-MIB-JUNIPER.mib
CISCO-IPSEC-POLICY-MAP-MIB.mib	HUAWEI-MEMORY-MIB.mib	OSPFV3-MIB.mib
CISCO-IPSLA-AUTOMEASURE-MIB.mib	HUAWEI-MFLP-MIB.mib	P-BRIDGE-MIB.mib
CISCO-IPSLA-ECHO-MIB.mib	HUAWEI-MIB.mib	PIM-MIB.mib
CISCO-IPSLA-JITTER-MIB.mib	HUAWEI-MINM-MIB.mib	PIM-STD-MIB.mib
CISCO-IPSLA-TC-MIB.mib	HUAWEI-MIRROR-MIB.mib	POWER-ETHERNET-MIB.mib
CISCO-ISDN-MIB.mib	HUAWEI-MPLSLSR-EXT-MIB.mib	PPP-IP-NCP-MIB.mib
CISCO-LICENSE-MGMT-MIB.mib	HUAWEI-MPLSOAM-MIB.mib	PPP-LCP-MIB.mib
CISCO-LOCAL-AUTH-USER-MIB.mib	HUAWEI-MPLSOAM-PS-MIB.mib	PPVPN-TC-MIB.mib
CISCO-LWAPP-AAA-MIB.mib	HUAWEI-MSTP-MIB.mib	PTOPO-MIB.mib
CISCO-LWAPP-AP-MIB.mib	HUAWEI-NAT-EXT-MIB.mib	PerfHist-TC-MIB.mib
CISCO-LWAPP-CCX-RM-MIB.mib	HUAWEI-OSPFV2-MIB.mib	Q-BRIDGE-MIB.mib
CISCO-LWAPP-CDP-MIB.mib	HUAWEI-OSPFV3-MIB.mib	RADIUS-ACC-CLIENT-MIB.mib

CISCO-LWAPP-CLIENT-ROAMING-CAPABILITY.mib	HUAWEI-PERFORMANCE-MIB.mib	RADIUS-AUTH-CLIENT-MIB.mib
CISCO-LWAPP-CLIENT-ROAMING-MIB.mib	HUAWEI-PIM-BSR-MIB.mib	RFC-1212.mib
CISCO-LWAPP-DHCP-MIB.mib	HUAWEI-PIM-STD-MIB.mib	RFC-1215.mib
CISCO-LWAPP-DOT11-CLIENT-CALIB-MIB.mib	HUAWEI-PORTAL-MIB.mib	RFC1155-SMI.mib
CISCO-LWAPP-DOT11-CLIENT-CCX-TC-MIB.mib	HUAWEI-PORTBLOCK-MIB.mib	RFC1213-MIB.mib
CISCO-LWAPP-DOT11-LDAP-MIB.mib	HUAWEI-PPP-MIB.mib	RFC1315-MIB.mib
CISCO-LWAPP-DOT11-MIB.mib	HUAWEI-RANAPS-MIB.mib	RFC1398-MIB.mib
CISCO-LWAPP-DOWNLOAD-MIB.mib	HUAWEI-RIPV2-EXT-MIB.mib	RIPv2-MIB.mib
CISCO-LWAPP-IDS-MIB.mib	HUAWEI-RM-EXT-MIB.mib	RMON-MIB.mib
CISCO-LWAPP-INTERFACE-MIB.mib	HUAWEI-RRPP-MIB.mib	RMON2-MIB.mib
CISCO-LWAPP-IPS-MIB.mib	HUAWEI-SECURITY-MIB.mib	RSTP-MIB.mib
CISCO-LWAPP-LINKTEST-MIB.mib	HUAWEI-SLB-MIB.mib	RSVP-MIB.mib
CISCO-LWAPP-LOCAL-AUTH-MIB.mib	HUAWEI-SNMP-EXT-MIB.mib	SMON-MIB.mib
CISCO-LWAPP-MDNS-MIB.mib	HUAWEI-SSH-MIB.mib	SNA-SDLC-MIB.mib
CISCO-LWAPP-MESH-BATTERY-MIB.mib	HUAWEI-STACK-MIB.mib	SNMP-COMMUNITY-MIB.mib
CISCO-LWAPP-MESH-LINKTEST-MIB.mib	HUAWEI-SYS-MAN-MIB.mib	SNMP-FRAMEWORK-MIB.mib
CISCO-LWAPP-MOBILITY-EXT-MIB.mib	HUAWEI-TAD-MIB.mib	SNMP-MPD-MIB.mib
CISCO-LWAPP-MOBILITY-MIB.mib	HUAWEI-TASK-MIB.mib	SNMP-NOTIFICATION-MIB.mib
CISCO-LWAPP-NETFLOW-MIB.mib	HUAWEI-TC-MIB.mib	SNMP-PROXY-MIB.mib
CISCO-LWAPP-REAP-MIB.mib	HUAWEI-TCP-MIB.mib	SNMP-REPEATER-MIB.mib
CISCO-LWAPP-RF-MIB.mib	HUAWEI-TRANSPARENTBRIDGE-MIB.mib	SNMP-TARGET-MIB.mib
CISCO-LWAPP-SI-MIB.mib	HUAWEI-TRNG-MIB.mib	SNMP-USER-BASED-SM-MIB.mib
CISCO-LWAPP-TC-MIB.mib	HUAWEI-TUNNEL-MIB.mib	SNMP-USM-AES-MIB.mib
CISCO-LWAPP-TRUSTSEC-MIB.mib	HUAWEI-USERLOG-MIB.mib	SNMP-USM-DH-OBJECTS-MIB.mib

CISCO-LWAPP-TSM-MIB.mib	HUAWEI-VE-MIB.mib	SNMP-VIEW-BASED-ACM-MIB.mib
CISCO-LWAPP-WLAN-MIB.mib	HUAWEI-VGMP-MIB.mib	SNMPv2-CONF.mib
CISCO-LWAPP-WLAN-SECURITY-MIB.mib	HUAWEI-VLL-STATISTIC-MIB.mib	SNMPv2-MIB.mib
CISCO-MEDIA-GATEWAY-MIB.mib	HUAWEI-VPLS-EXT-MIB.mib	SNMPv2-SMI.mib
CISCO-MOTION-MIB.mib	HUAWEI-VPLS-TNL-MIB.mib	SNMPv2-TC-v1.mib
CISCO-MPLS-LSR-EXT-STD-MIB.mib	HUAWEI-WAN-MIB.mib	SNMPv2-TC.mib
CISCO-MPLS-TC-EXT-STD-MIB.mib	HUAWEI-WLAN-DEVICE-MIB.mib	SNMPv2-TM.mib
CISCO-MPLS-TE-STD-EXT-MIB.mib	HUAWEI-WLAN-MIB.mib	SONET-MIB.mib
CISCO-NAC-TC-MIB.mib	HUAWEI-WLAN-QOS-MIB.mib	SYSAPPL-MIB.mib
CISCO-NBAR-PROTOCOL-DISCOVERY-MIB.mib	HUAWEI-WLAN-RADIO-MIB.mib	TCP-MIB.mib
CISCO-NETSYNC-MIB.mib	HUAWEI-WLAN-SERVICE-MIB.mib	TIMETRA-ALARM-MIB.mib
CISCO-NTP-MIB.mib	HUAWEI-WLAN-SYS-MIB.mib	TIMETRA-APS-MIB.mib
CISCO-OSPF-MIB.mib	HUAWEI-WLAN-UPDATE-MIB.mib	TIMETRA-ATM-MIB.mib
CISCO-OSPF-TRAP-MIB.mib	IANA-ADDRESS-FAMILY-NUMBERS-MIB.mib	TIMETRA-BGP-MIB.mib
CISCO-OTN-IF-MIB.mib	IANA-GMPLS-TC-MIB.mib	TIMETRA-CFLOWD-MIB.mib
CISCO-PAE-MIB.mib	IANA-ITU-ALARM-TC-MIB.mib	TIMETRA-CHASSIS-MIB.mib
CISCO-PAGP-MIB.mib	IANA-LANGUAGE-MIB.mib	TIMETRA-CLEAR-MIB.mib
CISCO-PIM-MIB.mib	IANA-RTPROTO-MIB.mib	TIMETRA-CONN-PROF-MIB.mib
CISCO-PING-MIB.mib	IANAifType-MIB.mib	TIMETRA-DIAMETER-MIB.mib
CISCO-POLICY-GROUP-MIB.mib	IEEE8021-CFM-MIB.mib	TIMETRA-DOT3-OAM-MIB.mib
CISCO-POWER-ETHERNET-EXT-MIB.mib	IEEE8021-PAE-MIB.mib	TIMETRA-FILTER-MIB.mib
CISCO-PRIVATE-VLAN-MIB.mib	IEEE8021-TC-MIB.mib	TIMETRA-GLOBAL-MIB.mib
CISCO-PROCESS-MIB.mib	IEEE802171-CFM-MIB.mib	TIMETRA-IPSEC-STATIC-SA-MIB.mib
CISCO-PRODUCTS-MIB.mib	IEEE8023-LAG-MIB.mib	TIMETRA-LLDP-MIB.mib
CISCO-PTP-MIB.mib	IEEE802dot11-MIB.mib	TIMETRA-LOG-MIB.mib
CISCO-RADIUS-EXT-MIB.mib	IF-INVERTED-STACK-MIB.mib	TIMETRA-MSDP-MIB.mib
CISCO-RF-MIB.mib	IF-MIB.mib	TIMETRA-OTU-MIB.mib

CISCO-RF-SUPPLEMENTAL-MIB.mib	IGMP-STD-MIB.mib	TIMETRA-ROUTE-POLICY-MIB.mib
CISCO-RTTMON-TC-MIB.mib	INET-ADDRESS-MIB.mib	TIMETRA-SYSTEM-MIB.mib
CISCO-SELECTIVE-VRF-DOWNLOAD-MIB.mib	INT-SERV-MIB.mib	TIMETRA-TC-MIB.mib
CISCO-SESS-BORDER-CTRLR-CALL-STATS-MIB.mib	INTEGRATED-SERVICES-MIB.mib	TIMETRA-TWAMP-MIB.mib
CISCO-SESS-BORDER-CTRLR-EVENT-MIB.mib	IP-FORWARD-MIB.mib	TIMETRA-VRRP-V3-MIB.mib
CISCO-SESS-BORDER-CTRLR-STATS-MIB.mib	IP-MIB.mib	TOKEN-RING-RMON-MIB.mib
CISCO-SMI.mib	IPMCAST-MIB.mib	TOKENRING-MIB.mib
CISCO-SONET-MIB.mib	IPMROUTE-MIB.mib	TRANSPORT-ADDRESS-MIB.mib
CISCO-ST-TC.mib	IPMROUTE-STD-MIB.mib	TUNNEL-MIB.mib
CISCO-STACKWISE-MIB.mib	IPV6-FLOW-LABEL-MIB.mib	UCD-DEMO-MIB.mib
CISCO-STP-EXTENSIONS-MIB.mib	IPV6-ICMP-MIB.mib	UCD-DISKIO-MIB.mib
CISCO-SUBSCRIBER-IDENTITY-TC-MIB.mib	IPV6-MIB.mib	UCD-DLMOD-MIB.mib
CISCO-SUBSCRIBER-SESSION-MIB.mib	IPV6-MLD-MIB.mib	UCD-IPFWACC-MIB.mib
CISCO-SUBSCRIBER-SESSION-TC-MIB.mib	IPV6-TC.mib	UCD-SNMP-MIB.mib
CISCO-SYSLOG-MIB.mib	IPV6-TCP-MIB.mib	UDP-MIB.mib
CISCO-SYSTEM-EXT-MIB.mib	IPV6-UDP-MIB.mib	VPLS-BGP-DRAFT-01-MIB.mib
CISCO-SYSTEM-MIB.mib	ISDN-MIB.mib	VPLS-GENERIC-DRAFT-01-MIB.mib
CISCO-TAP2-MIB.mib	ISIS-MIB.mib	VPLS-LDP-DRAFT-01-MIB.mib
CISCO-TC.mib	ITU-ALARM-MIB.mib	VPN-TC-STD-MIB.mib
CISCO-TCP-MIB.mib	ITU-ALARM-TC-MIB.mib	VRRP-MIB.mib
CISCO-TEMP-LWAPP-DHCP-MIB.mib	JNX-IPSEC-MONITOR-MIB.mib	ietf-inet-types.mib
CISCO-TRUSTSEC-SXP-MIB.mib	JNX-L2TP-MIB.mib	ietf-yang-smiv2.mib
CISCO-TRUSTSEC-TC-MIB.mib	JUNIPER-ALARM-MIB.mib	ietf-yang-types.mib

List of Pre-loaded YANG Modules for MDT Collection

This section lists the YANG modules that the Collection Service supports for MDT collection on Cisco IOS XR devices.

Table 2: Supported YANG Modules

Cisco-IOS-XR-Subscriber-infra-subdb-oper
Cisco-IOS-XR-aaa-locald-oper
Cisco-IOS-XR-aaa-nacm-oper
Cisco-IOS-XR-aaa-protocol-radius-oper
Cisco-IOS-XR-alarmgr-server-oper
Cisco-IOS-XR-ascii-ltrace-oper
Cisco-IOS-XR-asic-errors-ope
Cisco-IOS-XR-atm-vcn-oper
Cisco-IOS-XR-bundlemgr-oper
Cisco-IOS-XR-cdp-oper
Cisco-IOS-XR-clns-isis-oper
Cisco-IOS-XR-cmproxy-oper
Cisco-IOS-XR-cofo-infra-oper
Cisco-IOS-XR-config-cfgmgr-exec-oper
Cisco-IOS-XR-config-cfgmgr-oper
Cisco-IOS-XR-controller-optics-oper
Cisco-IOS-XR-controller-otu-oper
Cisco-IOS-XR-crypto-macsec-mka-oper
Cisco-IOS-XR-crypto-macsec-secy-oper
Cisco-IOS-XR-crypto-sam-oper
Cisco-IOS-XR-crypto-ssh-oper
Cisco-IOS-XR-dot1x-oper
Cisco-IOS-XR-drivers-media-eth-oper
Cisco-IOS-XR-dwdm-ui-oper
Cisco-IOS-XR-es-acl-oper
Cisco-IOS-XR-ethernet-lldp-oper
Cisco-IOS-XR-evpn-oper
Cisco-IOS-XR-fib-common-oper

Cisco-IOS-XR-flowspec-oper
Cisco-IOS-XR-freqsync-oper
Cisco-IOS-XR-ha-eem-policy-oper
Cisco-IOS-XR-icpe-infra-oper
Cisco-IOS-XR-iedge4710-oper
Cisco-IOS-XR-ifmgr-oper
Cisco-IOS-XR-infra-alarm-logger-oper
Cisco-IOS-XR-infra-correlator-oper
Cisco-IOS-XR-infra-objmgr-oper
Cisco-IOS-XR-infra-policymgr-oper
Cisco-IOS-XR-infra-rcmd-oper
Cisco-IOS-XR-infra-rmf-oper
Cisco-IOS-XR-infra-rsi-oper
Cisco-IOS-XR-infra-serg-oper
Cisco-IOS-XR-infra-statsd-oper
Cisco-IOS-XR-infra-syslog-oper
Cisco-IOS-XR-infra-tc-oper
Cisco-IOS-XR-infra-xtc-agent-oper
Cisco-IOS-XR-infra-xtc-oper
Cisco-IOS-XR-invmgr-oper
Cisco-IOS-XR-ip-bfd-oper
Cisco-IOS-XR-ip-daps-oper
Cisco-IOS-XR-ip-domain-oper
Cisco-IOS-XR-ip-iarm-v4-oper
Cisco-IOS-XR-ip-iarm-v6-oper
Cisco-IOS-XR-ip-iep-oper
Cisco-IOS-XR-ip-mobileip-oper
Cisco-IOS-XR-ip-ntp-admin-oper
Cisco-IOS-XR-ip-ntp-oper
Cisco-IOS-XR-ip-pfilter-oper
Cisco-IOS-XR-ip-rib-ipv4-oper
Cisco-IOS-XR-ip-rib-ipv6-oper
Cisco-IOS-XR-ip-rip-oper

Cisco-IOS-XR-ip-rsvp-oper
Cisco-IOS-XR-ip-sbfd-oper
Cisco-IOS-XR-ip-tcp-oper
Cisco-IOS-XR-ip-udp-oper
Cisco-IOS-XR-ipv4-acl-oper
Cisco-IOS-XR-ipv4-arp-oper
Cisco-IOS-XR-ipv4-autorp-oper
Cisco-IOS-XR-ipv4-bgp-oc-oper
Cisco-IOS-XR-ipv4-bgp-oper
Cisco-IOS-XR-ipv4-dhcpd-oper
Cisco-IOS-XR-ipv4-hsrp-oper
Cisco-IOS-XR-ipv4-igmp-oper
Cisco-IOS-XR-ipv4-io-oper
Cisco-IOS-XR-ipv4-ospf-oper
Cisco-IOS-XR-ipv4-pim-oper
Cisco-IOS-XR-ipv4-vrrp-oper
Cisco-IOS-XR-ipv6-acl-oper
Cisco-IOS-XR-ipv6-io-oper
Cisco-IOS-XR-ipv6-ma-oper
Cisco-IOS-XR-ipv6-nd-oper
Cisco-IOS-XR-ipv6-new-dhcpv6d-oper
Cisco-IOS-XR-ipv6-ospfv3-oper
Cisco-IOS-XR-l2-eth-infra-oper
Cisco-IOS-XR-l2rib-oper
Cisco-IOS-XR-l2vpn-oper
Cisco-IOS-XR-lib-keychain-oper
Cisco-IOS-XR-lib-mpp-oper
Cisco-IOS-XR-linux-os-reboot-history-oper
Cisco-IOS-XR-lmp-oper
Cisco-IOS-XR-lpts-ifib-oper
Cisco-IOS-XR-lpts-pa-oper
Cisco-IOS-XR-lpts-pre-ifib-oper
Cisco-IOS-XR-macsec-ctrlr-oper

Cisco-IOS-XR-man-ems-oper
Cisco-IOS-XR-man-ipsla-oper
Cisco-IOS-XR-man-xml-ttyagent-oper
Cisco-IOS-XR-manageability-object-tracking-oper
Cisco-IOS-XR-manageability-perfmgmt-oper
Cisco-IOS-XR-mediasvr-linux-oper
Cisco-IOS-XR-mpls-io-oper
Cisco-IOS-XR-mpls-ldp-oper
Cisco-IOS-XR-mpls-ldp-oper
Cisco-IOS-XR-mpls-ldp-oper
Cisco-IOS-XR-mpls-oam-oper
Cisco-IOS-XR-mpls-static-oper
Cisco-IOS-XR-mpls-te-oper
Cisco-IOS-XR-mpls-vpn-oper
Cisco-IOS-XR-ncs6k-acl-oper
Cisco-IOS-XR-nto-misc-oper
Cisco-IOS-XR-nto-misc-shmem-oper
Cisco-IOS-XR-nto-misc-shprocmem-oper
Cisco-IOS-XR-ocni-intfbase-oper
Cisco-IOS-XR-ocni-local-routing-oper
Cisco-IOS-XR-ocni-oper
Cisco-IOS-XR-pbr-oper
Cisco-IOS-XR-pbr-vservice-ea-oper
Cisco-IOS-XR-pbr-vservice-mgr-oper
Cisco-IOS-XR-perf-meas-oper
Cisco-IOS-XR-pfi-im-cmd-ctrlr-oper
Cisco-IOS-XR-pfi-im-cmd-oper
Cisco-IOS-XR-plat-chas-invmgr-ng-oper
Cisco-IOS-XR-plat-chas-invmgr-oper
Cisco-IOS-XR-pmengine-oper
Cisco-IOS-XR-policy-repository-oper
Cisco-IOS-XR-ppp-ea-oper
Cisco-IOS-XR-ppp-ma-oper
Cisco-IOS-XR-pppoe-ea-oper

Cisco-IOS-XR-procmem-oper
Cisco-IOS-XR-ptp-oper
Cisco-IOS-XR-qos-ma-oper
Cisco-IOS-XR-rgmgr-oper
Cisco-IOS-XR-sdr-invmgr-diag-oper
Cisco-IOS-XR-segment-routing-ms-oper
Cisco-IOS-XR-shellutil-filesystem-oper
Cisco-IOS-XR-shellutil-oper
Cisco-IOS-XR-show-fpd-loc-ng-oper
Cisco-IOS-XR-snmp-agent-oper
Cisco-IOS-XR-spirit-install-instmgr-oper
Cisco-IOS-XR-sse-span-oper
Cisco-IOS-XR-subscriber-accounting-oper
Cisco-IOS-XR-subscriber-ipsub-oper
Cisco-IOS-XR-subscriber-pppoe-ma-oper
Cisco-IOS-XR-subscriber-session-mon-oper
Cisco-IOS-XR-subscriber-srg-oper
Cisco-IOS-XR-syncc-oper
Cisco-IOS-XR-telemetry-model-driven-oper
Cisco-IOS-XR-tty-management-cmd-oper
Cisco-IOS-XR-tty-server-oper
Cisco-IOS-XR-tunnel-l2tun-oper
Cisco-IOS-XR-tunnel-nve-oper
Cisco-IOS-XR-tunnel-vpdn-oper
Cisco-IOS-XR-upgrade-fpd-oper
Cisco-IOS-XR-wanphy-ui-oper
Cisco-IOS-XR-wd-oper
Cisco-IOS-XR-wdsysmon-fd-oper
cisco-xr-ietf-netconf-acm-deviations
cisco-xr-ietf-netconf-monitoring-deviations
cisco-xr-openconfig-acl-deviations
cisco-xr-openconfig-bgp-policy-deviations
cisco-xr-openconfig-if-ethernet-deviations

cisco-xr-openconfig-if-ip-deviations
cisco-xr-openconfig-interfaces-deviations
cisco-xr-openconfig-lacp-deviations
cisco-xr-openconfig-lldp-deviations
cisco-xr-openconfig-local-routing-deviations
cisco-xr-openconfig-network-instance-deviations
cisco-xr-openconfig-optical-amplifier-deviations
cisco-xr-openconfig-platform-deviations
cisco-xr-openconfig-platform-transceiver-deviations
cisco-xr-openconfig-rib-bgp-deviations
cisco-xr-openconfig-routing-policy-deviations
cisco-xr-openconfig-rsvp-sr-ext-deviations
cisco-xr-openconfig-telemetry-deviations
cisco-xr-openconfig-terminal-device-deviations
cisco-xr-openconfig-transport-line-protection-deviations
cisco-xr-openconfig-vlan-deviations
ietf-netconf
ietf-netconf-acm
ietf-netconf-monitoring
openconfig-acl
openconfig-aft
openconfig-aft-common
openconfig-aft-ethernet
openconfig-aft-ipv4
openconfig-aft-ipv6
openconfig-aft-mpls
openconfig-aft-network-instance
openconfig-aft-pf
openconfig-bgp
openconfig-bgp-common
openconfig-bgp-common-multiprotocol
openconfig-bgp-common-structure
openconfig-bgp-global

openconfig-bgp-neighbor
openconfig-bgp-peer-group
openconfig-bgp-policy
openconfig-channel-monitor
openconfig-extensions
openconfig-if-aggregate
openconfig-if-ethernet
openconfig-if-ip
openconfig-interfaces
openconfig-isis
openconfig-isis-lsp
openconfig-isis-policy
openconfig-isis-routing
openconfig-lacp
openconfig-ldp
openconfig-local-routing
openconfig-mpls
openconfig-mpls-igp
openconfig-mpls-ldp
openconfig-mpls-rsvp
openconfig-mpls-sr
openconfig-mpls-static
openconfig-mpls-te
openconfig-network-instance
openconfig-network-instance-l2
openconfig-network-instance-l3
openconfig-optical-amplifier
openconfig-packet-match
openconfig-platform
openconfig-platform-transceiver
openconfig-rib-bgp
openconfig-routing-policy
openconfig-rsvp-sr-ext

openconfig-segment-routing
openconfig-telemetry
openconfig-terminal-device
openconfig-transport-line-common
openconfig-transport-line-protection
openconfig-vlan

