

排除Cisco IOS XR的XML故障

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[限制内存问题](#)

[进一步的问题描述](#)

[解决方案](#)

[思科IOS XR中的XML](#)

[使用XML API](#)

[XML API支持的操作](#)

[提供的XML架构服务](#)

[生成查询](#)

[测试XML查询](#)

[思科IOS XR中的XML深入探讨](#)

[与客户端的API交互](#)

[Cisco IOS XR中的XML错误](#)

[查询出错时](#)

[错误类型](#)

[其他建议](#)

简介

本文档介绍在服务请求和思科工具中看到的XML内存限制问题。

先决条件

要求

Cisco 建议您了解以下主题：

- Cisco IOS® XR平台
- XML (可扩展标记语言) 基础设施
- 公共对象请求代理体系结构(CORBA)
- 元件管理系统(EMS)
- 外部数据管理器(EDM)

使用的组件

本文档中的信息基于ASR9000。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

限制内存问题

我们打开一个案例，询问以下问题：

1. 路由器上可配置的最大内存是多少？
2. 是否有办法中断大量XML回复？

回答：

1. 最大内存根据RSP/节点运行的版本而变化（cXR和eXR的版本）

在64位版本(eXR)中。您有一个限制内存大小：

```
RP/0/RSP1/CPU0:XR#show version
Wed Jul 26 21:10:16.761 IST
Cisco IOS XR Software, Version 7.1.3
Copyright (c) 2013-2020 by Cisco Systems, Inc.
```

Build Information:

```
Built By      : gopalk2
Built On     : Thu Nov 26 10:51:48 PST 2020
Built Host   : iox-ucs-027
Workspace    : /auto/srcarchive17/prod/7.1.3/asr9k-x64/ws
Version      : 7.1.3
Location     : /opt/cisco/XR/packages/
Label       : 7.1.3
```

```
cisco ASR9K () processor
System uptime is 2 weeks 17 hours 22 minutes
```

```
RP/0/RSP1/CPU0:XR# configuration
RP/0/RSP1/CPU0:XR(config)#xml agent throttle ?
  memory      Memory usage
  process-rate Process rate
RP/0/RSP1/CPU0:XR(config)#xml agent throttle memory ?
<100-1024> Size of the memory usage in Mbytes per session (default 300 Mbytes)
```

32位版本(cXR)。您还有另一个限制：

```
RP/0/RSP0/CPU0:XR#show version
Cisco IOS XR Software, Version 6.4.2[Default]
Copyright (c) 2020 by Cisco Systems, Inc.
```

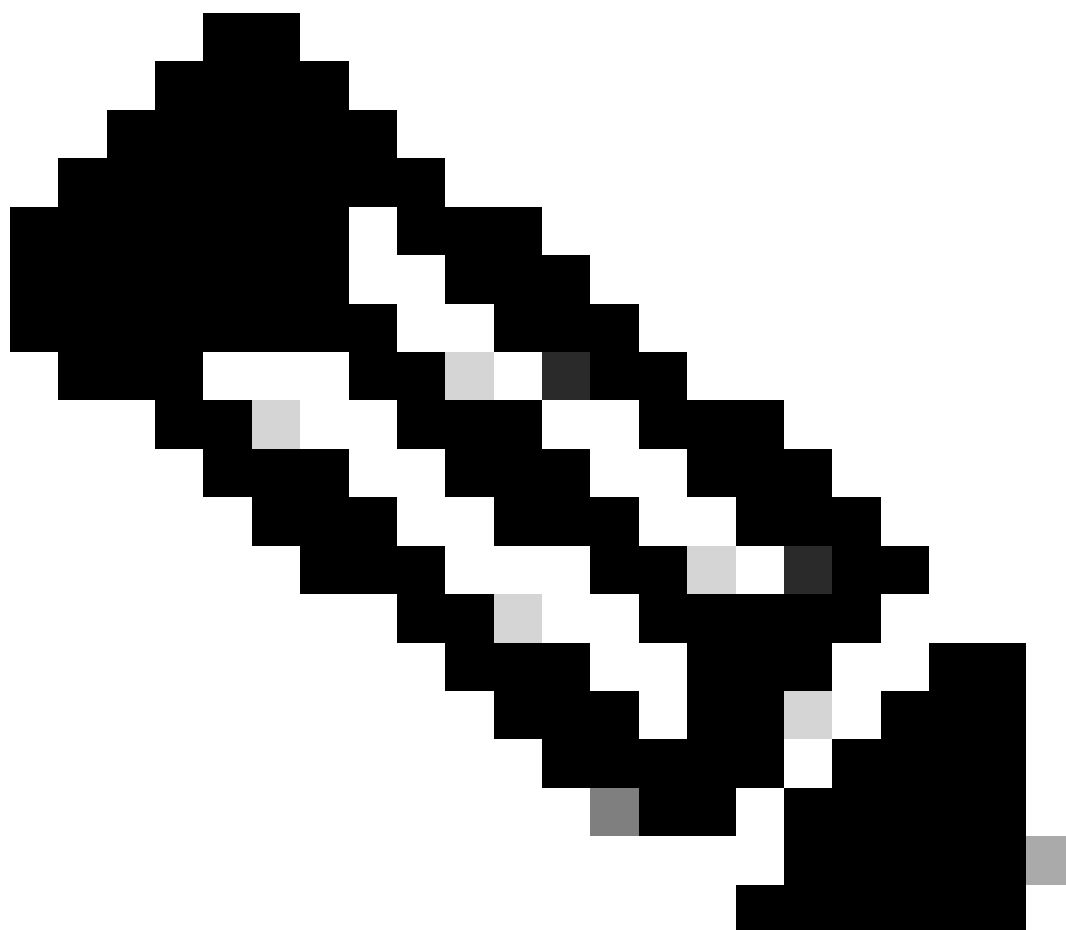
```
ROM: System Bootstrap, Version 10.59(c) 1994-2014 by Cisco Systems, Inc.
<snip>
```

Configuration register on node 0/RSP0/CPU0 is 0x102
Boot device on node 0/RSP0/CPU0 is disk0:

RP/0/RSP0/CPU0:XR# configuration

RP/0/RSP0/CPU0:XR(config)#xml agent throttle memory ?

<100-600> Size of the memory usage in Mbytes per session (default 300 Mbytes)



注：任一版本中的默认值均为300 MB

2. 有多种方法可以中断XML回复：

- 迭代器：

配置迭代器时，您将在特定窗口大小中对XML总响应进行分段。窗口具有迭代器大小。例如，如果响应为1 Gb，迭代器大小为500 Mb，则可以将XML分段为2个回复。

此方法通过添加（基于迭代器大小）GetNext操作（响应所需）来更改响应。

一个会话最多可以有10个迭代器。

- 限制 (进程速率) :

此功能限制了XML进程占用的内存。如果进程超过内存，它会回复错误消息“The throttle on the memory usage has been reached”。

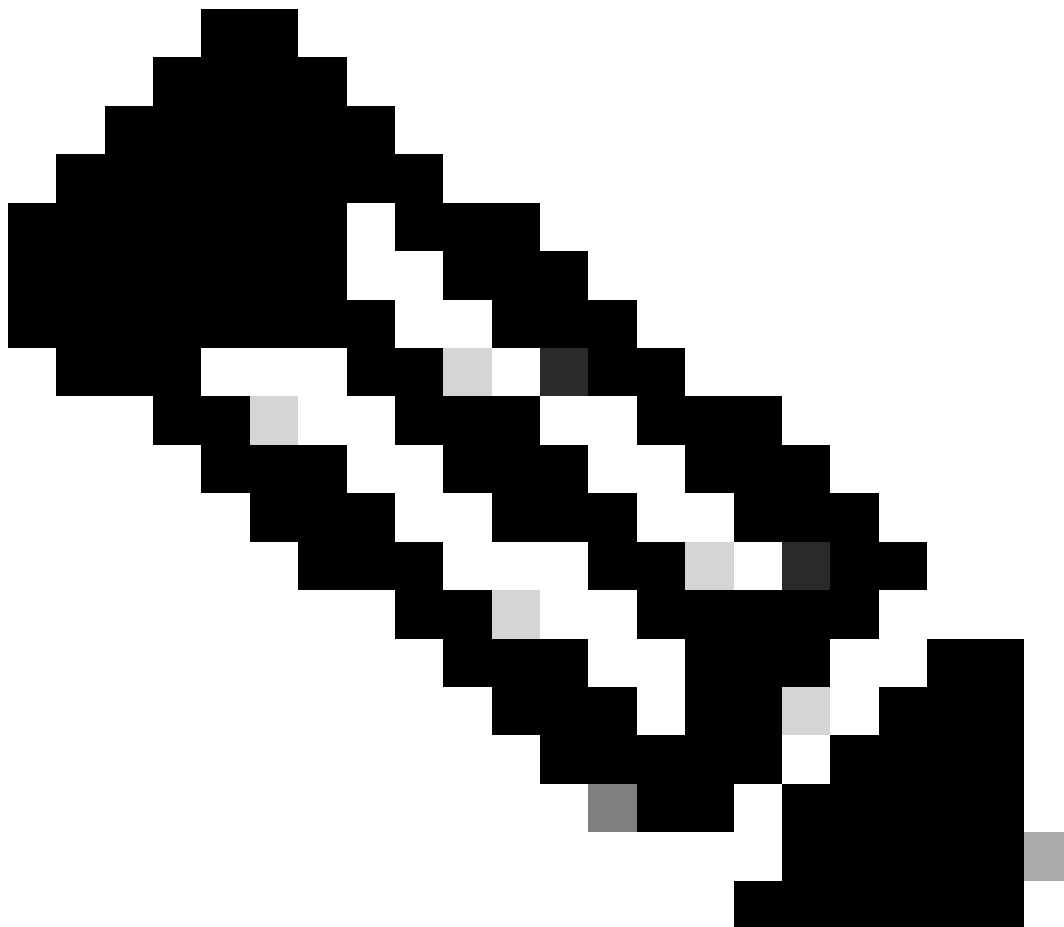
- 流传输 :

与迭代器类似，流式处理将XML响应分段到特定窗口。区别在于响应，它会删除GetNext操作和迭代器ID。XML发送流传输的数据包，当数据流结束时，客户端生成响应。

进一步的问题描述

对于需要自动化的使用案例，请使用工具[pyIOSXR](#)。此自动化工具是一种XML代理，它有助于发出一些show命令，并整体连接到设备。

每次通过此代理发送大请求时，都会显示错误：



注意：只有思科注册用户才能访问思科内部工具和信息。

RESPONSE ERROR: 0xa367a600 'XML Service Library' detected the 'fatal' condition 'The throttle on the me

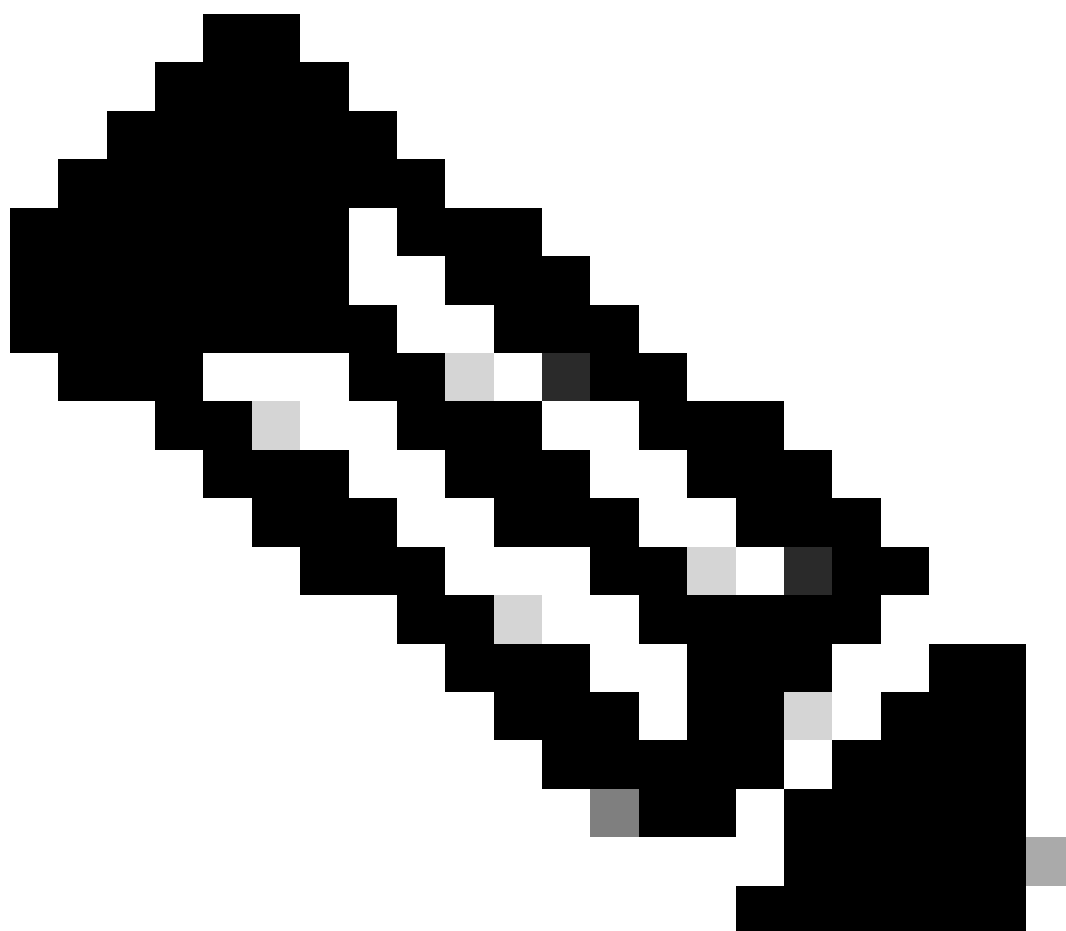
通过显示“优化查询较小数据的请求”的错误消息，您可以使用迭代器配置代理XML API。理论上，这允许对响应进行分段。

配置迭代器时，会显示相同的错误消息：“RESPONSE ERROR...Please optimize the request to query small data”。

当显示错误时，下一步是了解该功能（本例中为迭代器）不适用于查询的原因。

[pyIOSXR](#)建议在设备中正确启用XML代理，这意味着API不允许使用迭代器。

下一步是测试第二个选项：流传输。



注意：pyIOSXR不允许我们使用除具有下列元素的基本XML查询以外的其他标头：

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Operation>
  .
  .
  .
  </Operation>
</Request>
```

流和迭代器会在查询中添加标头。流添加流ID可帮助系统构建完整响应。迭代器添加GET_Next和迭代器ID。

测试流也显示了与Iterators相同的错误消息。

解决方案

在“对设备执行查询的更多建议”一节中，提供了更多工具来解决这些问题。通配符就是其中之一。通配符是内存限制限制的解决方案。

通配符可构建特定查询以避免请求不必要的信息。例如，对于BGP信息，请使用show route bgp命令，而不使用通用show route 命令。此示例适用于XML查询和逻辑。向系统请求批量信息可能会产生内存和处理问题。

打开案例时，使用以下查询：

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get>
    <Operational>
      <BGP MajorVersion="36" MinorVersion="1">
        <InstanceTable>
          <Instance>
            <Naming>
              <InstanceName>
                DEFAULT
              </InstanceName>
            </Naming>
            <InstanceActive>
              <VRFTTable>
                <VRF>
                  <Naming>
                    <VRFName>
                      VRF_NAME
                    </VRFName>
                  </Naming>
                <AFTTable>
                  <AF>
                    <Naming>
```

```

        <AFName>
            NAME
        </AFName>
    </Naming>
    <NetworkTable>

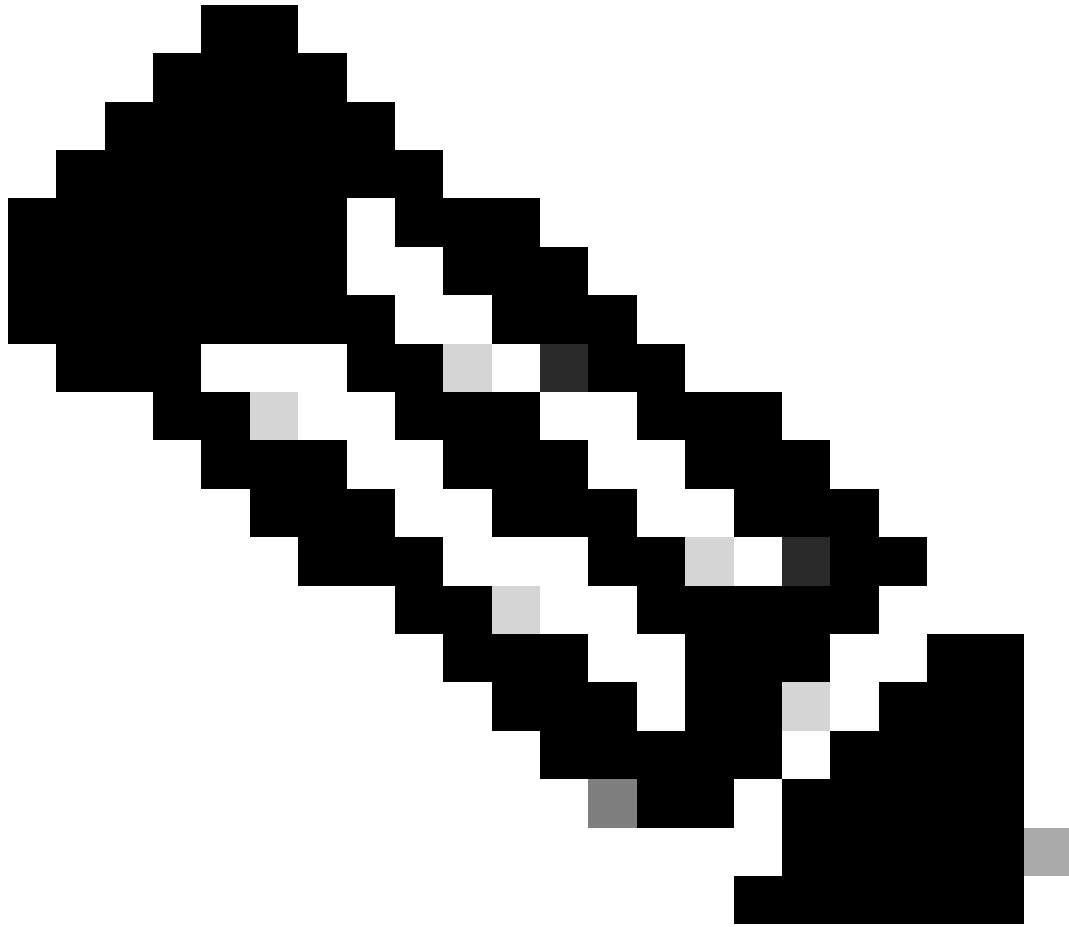
        </NetworkTable>
    </AF>
</AFTable>
</VRF>
</VRFTable>
</InstanceActive>
</Instance>
</InstanceTable>
</BGP>>
</Operational>
</Get>
</Request>

```

此查询针对完整的BGP表。对于此非特定请求，响应接近2.2 Gb，因此已达到内存限制限制。

要解决此问题，需要指定查询，这样系统就可以处理查询并返回信息。

思科IOS XR中的XML



注意：如果需要更多信息、规范或其他命令，请单击下一链接“XML编程指南：[XR XML编程指南](#)”

XML定义了如何显示和构建数据。这是一种分析计算机理解为位并展示结构化、标准化信息的方式。

XML具有以下结构：

```
<init>
  <body>
    <message>This is an example</message>
  </body>
</init>
```

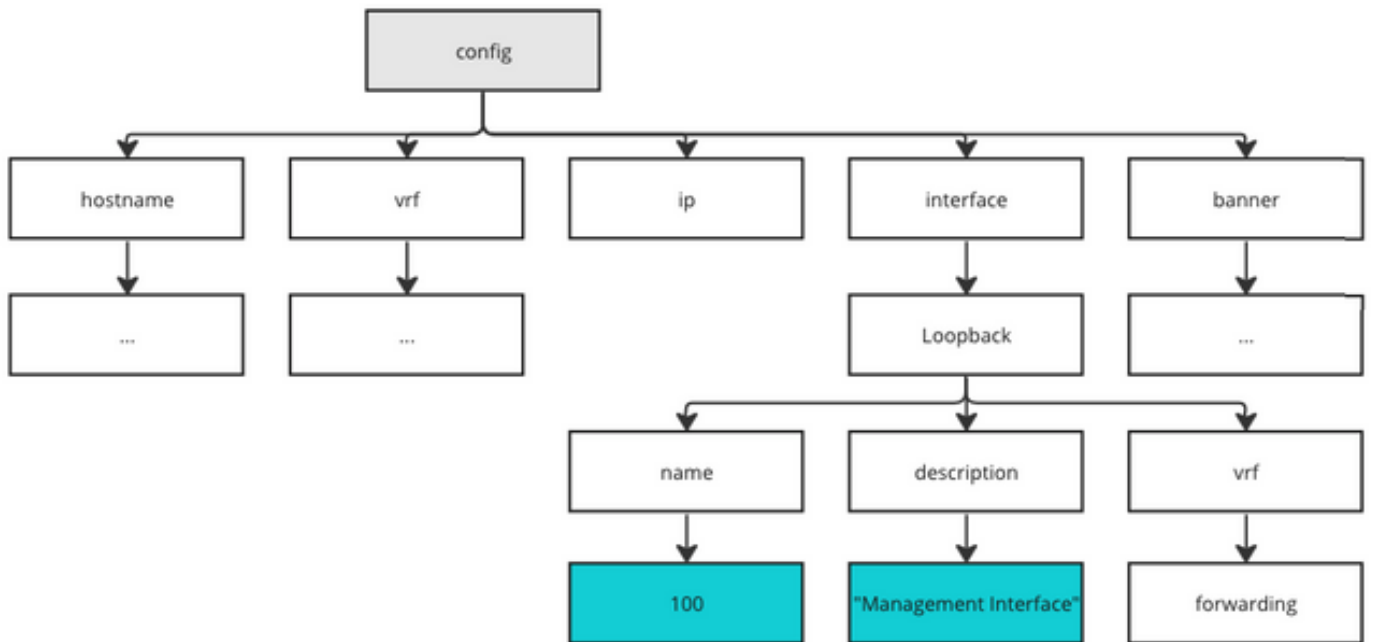
所有标记都包含两个部分，即开始标记(<init>)和结束标记(</init>)。如果未使用此结构，则XML无法理解标签的结束位置。

XML由两个实体组成：

- 容器
- 枝叶

此结构化数据类似于将XML数据作为树进行比较。容器是一个分支，每个分支都有树叶。枝叶不包含除信息以外的任何信息。

例如，下一棵树的根为灰色，容器为白色，枝叶为蓝色。

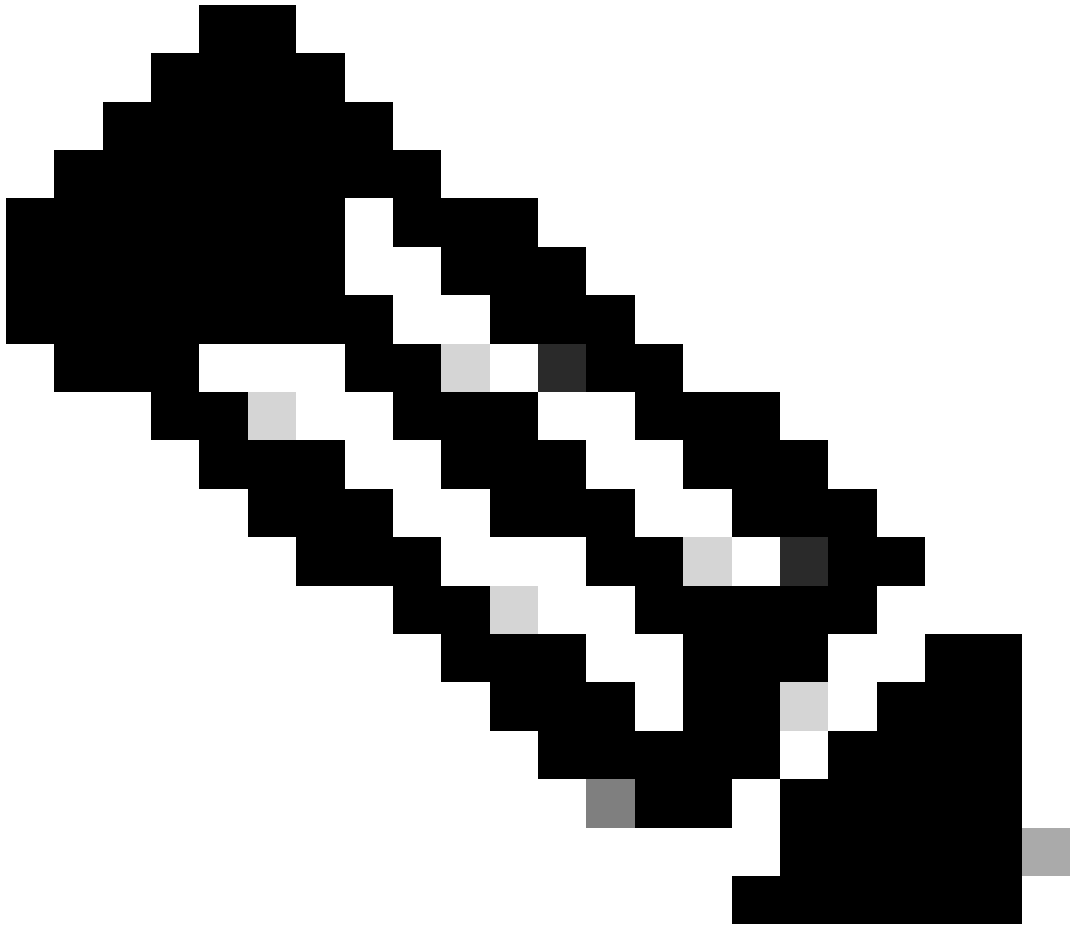


使用XML API

要测试/使用XML API，您首先需要的是查询。

1. 查询具有标题：

```
<?xml version="1.0" encoding="UTF-8"?>
```



注意：这是不需要结束标记的唯一标记。

2. 您需要为请求添加标记。在请求中，您需要指定版本。

```
<Request MajorVersion="1" MinorVersion="0">
```

3. 通过标题、请求和版本，正文继续执行API具有以下内容的任何XML操作：

```
<Operation>  
<Operation_1>  
<Operation_2>  
...  
<Operation_n>
```



注意：请注意，每个查询中的请求包含不同的操作。无需对每个操作执行请求。

下一个示例显示一个包含所有必需信息的请求：

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Operation></Operation>
  <Operation_1></Operation_1>
  <Operation_2></Operation_2>
  ...
  <Operation_n></Operation_n>
</Request>
```

XML API支持的操作

思科IOS XR支持5个操作，允许用户与XML架构具有的信息交互：

1. 本机数据操作：

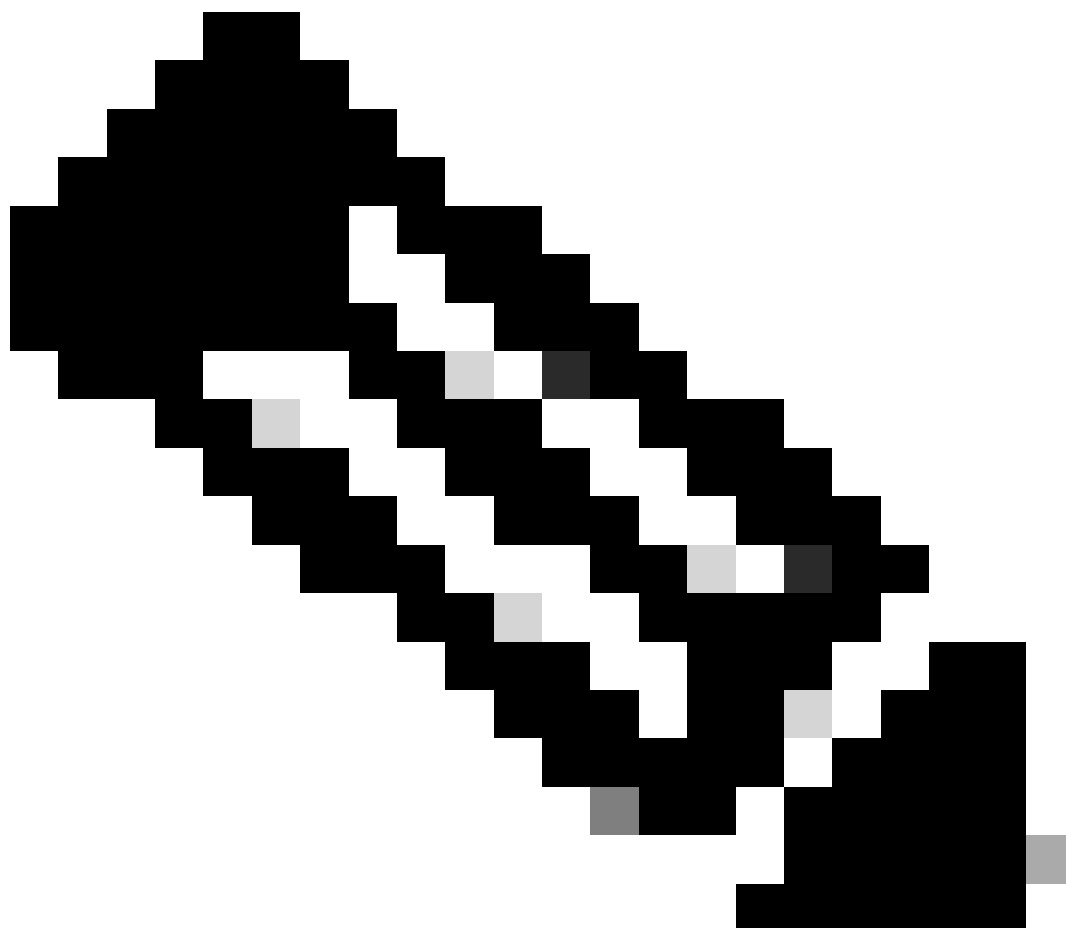
- `<Get></Get>`：收集行动数据项（枝叶拥有的数据）。
- `<Set></Set>`：配置、更改或添加数据项。
- `<Delete></Delete>`：删除一个或多个数据项。
- `<GetVersionInfo></GetVersionInfo>`：检索所请求组件的主版本号和次版本号的操作。
- `<GetDataSpaceInfo></GetDataSpaceInfo>`：此操作标记显示映射到容器的枝叶名称。

2. CLI操作：

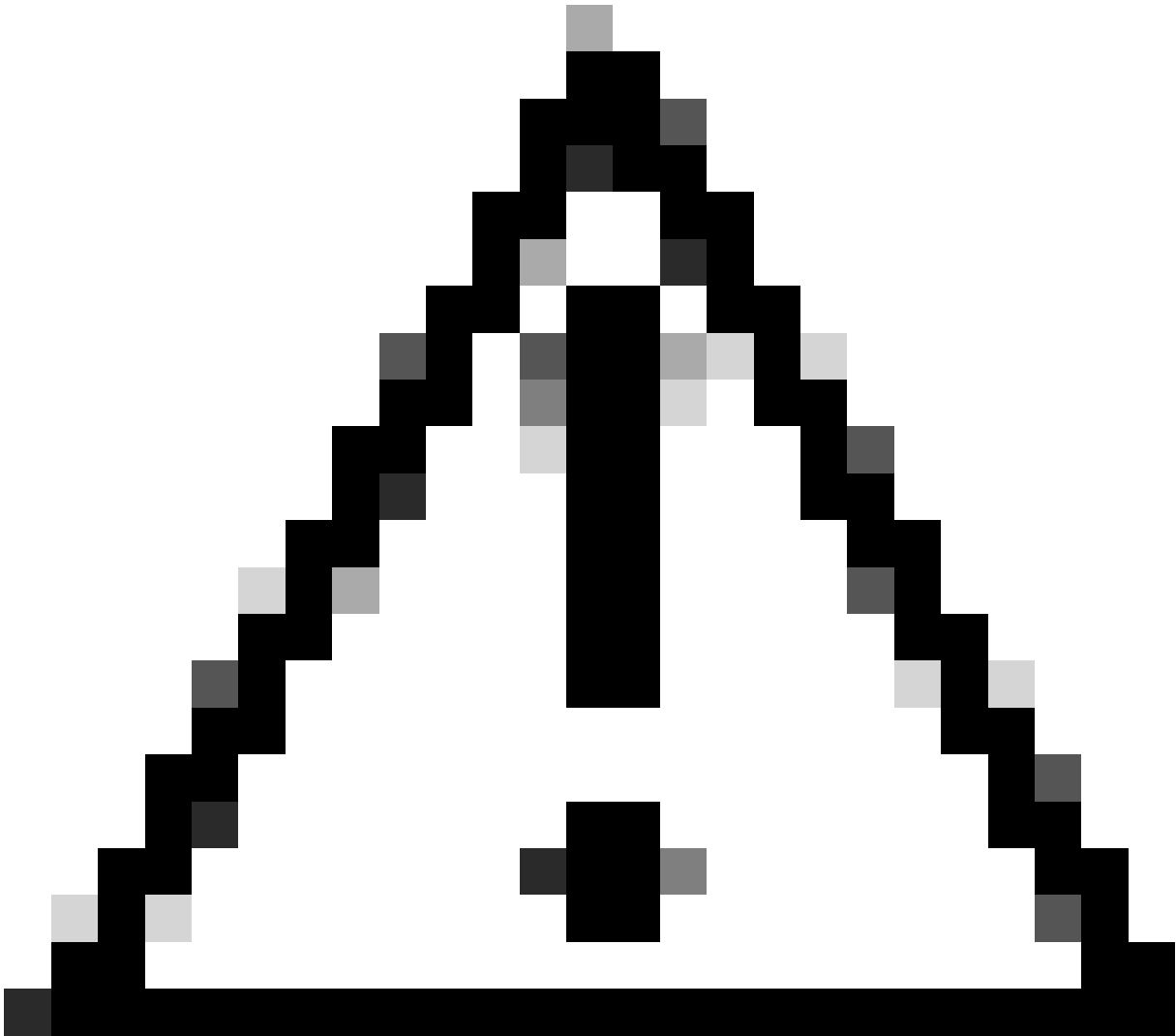
- `<CLI></CLI>`：允许我们发出配置请求作为CLI命令的标记。
- `<GetNext></GetNext>`：如果数据大于块大小，此标记会添加到下一个分段回复。

提供的XML架构服务

XML API实现以下命名空间：



注意：如果需要查看根对象，请访问：[Cisco IOS-XR XML网络管理](#)。



注意：请注意，每个命名空间分别监控不同的数据和操作。

-
- 配置：允许Get、Set和Delete操作的操作。
 - 操作：类似于CLI show命令。
 - 操作：支持设置操作。通过此操作，您可以访问操作数据，例如clear命令和其他类似命令。
 - AdminOperational：访问管理员操作数据。
 - AdminAction：访问管理操作数据。
 - AdminConfiguration：访问管理配置数据。

生成查询

有两种不同的旅行途径。为查询选择命名空间后，需要使用架构进行处理：

1. XML架构文档

这些选项提供显示容器和枝叶的完整XML树。对于文档，请单击[CRS XML方案](#)。

系统随即会显示文档页面。

XML Schema Documentation for the Cisco CRS Router, Cisco IOS XR Release 4.3.0

[HOME](#)
[SUPPORT](#)
[PRODUCT SUPPORT](#)
[ROUTERS](#)
[CISCO CARRIER ROUTING SYSTEM](#)
[Programming Guides](#)

XML Schema Object Documentation for root_objects, version 1. 0.

Objects

<Configuration>

Description: Root object for configuration

Child objects:

- [<AAA>](#) - Authentication, Authorization and Accounting
- [<AIB>](#) - AIB configuration data
- [<APS>](#) - SONET Automatic Protection Switching (APS) configuration
- [<ARP>](#) - ARP configuration
- [<ATM>](#) - Global ATM configuration
- [<ActiveNodeTable>](#) - Per-node configuration for active nodes
- [<BFD>](#) - BFD Configuration
- [<BGP>](#) - BGP configuration commands
- [<CDP>](#) - Global CDP configuration data
- [<CEF>](#) - Establish CEF configuration
- [<CallHome>](#) - Set CallHome parameters
- [<Clock>](#) - Configure time-of-day clock
- [<ControlPlane>](#) - Configure control Plane
- [<Crypto>](#) - Crypto configuration
- [<DHCPV6D>](#) - IPV6 DHCP config data
- [<DHCPv4>](#) - DHCP IPv4 configuration
- [<DynamicTemplate>](#) - All dynamic template configurations
- [<EIGRP>](#) - Configure Neighbor
- [<ERP>](#) - Ethernet Ring Protection
- [<ES_ACL>](#) - Layer 2 ACL configuration data
- [<ErrorDisable>](#) - Error Disable Configuration
- [<EthernetFeatures>](#) - Ethernet Features Configuration
- [<EventManager>](#) - Event manager configuration
- [<Exception>](#) - Core dump configuration commands
- [<FTP>](#) - FTP configuration
- [<GenericInterfaceListTable>](#) - Generic Interface List configuration
- [<GlobalAF>](#) - Default VRF configuration
- [<GlobalInterfaceConfiguration>](#) - Global scoped configuration for interfaces
- [<HSRP>](#) - HSRP configuration
- [<HSRPCfg>](#) - HSRP configuration

此页显示对象、容器和子项。每个对象都包含一个子对象。如果子对象不包含任何其他容器，则将其视为枝叶。

您可以单击子项，此时将显示用于生成查询的信息。例如，对于IPv4：

<IPV4>

Description: IPV4 related services

Task IDs required: ipv4

Parent objects:

[<Services>](#)

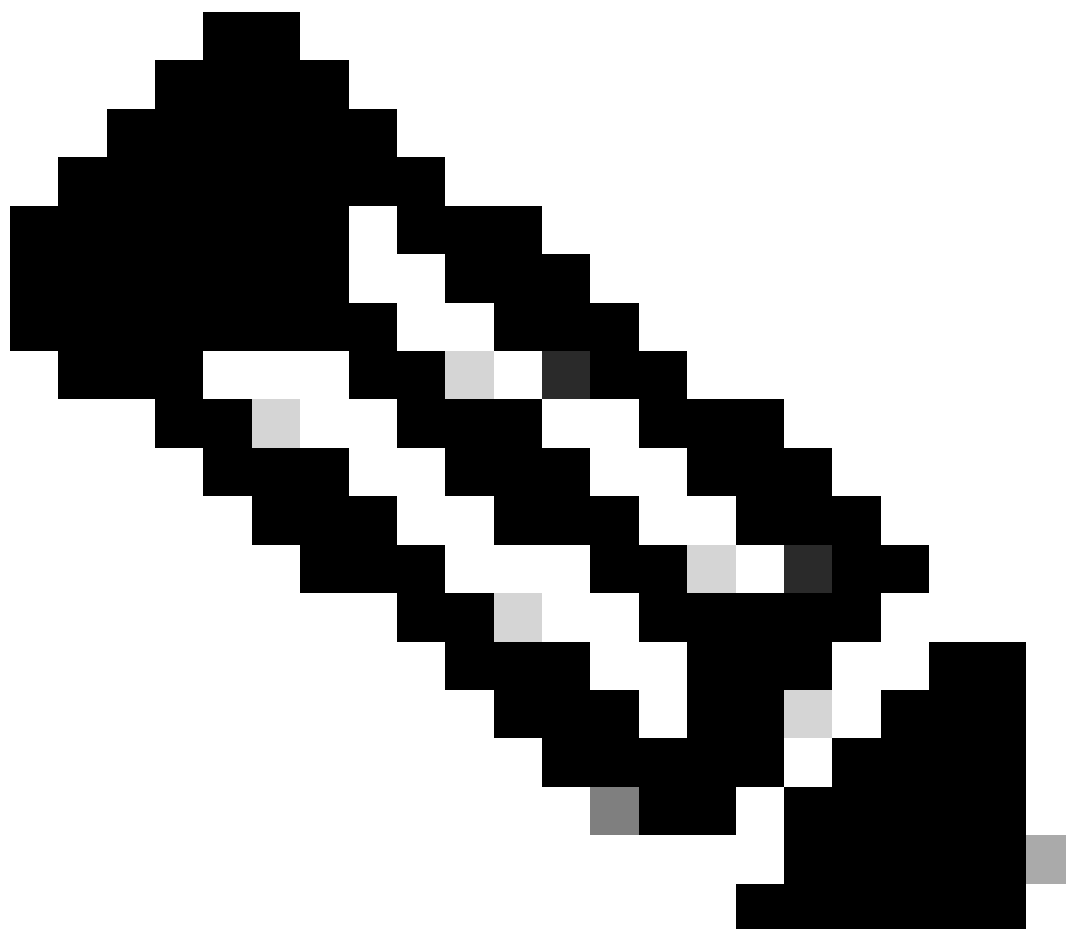
Child objects:

[<SmallServers>](#) - Describing IPV4 and IPV6 small servers

Available Paths:

- [<Configuration>](#) [<IP>](#) [<Cinetd>](#) [<Services>](#) [<IPV4>](#)

在前面的映像中，对于任何IPv4查询，都可以运行针对容器IP、容器Cinetd、容器Services和IPv4的配置操作。



注意：执行查询需要叶的任何操作。

查询是下一个查询：

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get>
    <Configurations>
      <IP>
        <Cinetd>
          <Services>
            <IPv4/>
          </Services>
        </Cinetd>
      </IP>
    </Configurations>
  </Get>
</Request>
```

```
</Get>
</Request>
```

2. XML API

XML API嵌入在思科IOS XRI中。若要查看XML树，请使用show xml schema命令。通过发出命令，整个XML架构在类似于Linux的CLI中可见。

在此CLI中，有以下操作：

? : 就像— 帮助

LS : 列出分支机构用户所在的容器/枝叶。

<operation> : 通过发出? 您可以检查是否执行了操作和发出了命令：

一些操作包括：

-config

- 管理配置

- cd

- 列表

- 操作

-动作

光盘：更改目录/容器。

示例显示以下内容：

```
RP/0/RP0/CPU0:XR#show xml schema
Username:admin
Password:
```

```
xml-schema[config]:> ?
```

```
config                oper                action
adminconfig          adminoper          adminaction
cd                   pwd                classinfo
list                 ls                 datalist
walk                 walkdata           get
hierarchy            quit               exit
help
```

```
xml-schema[config]:> ls
```

```
[container]          RIP
[container]          TCL
[container]          LawfulIntercept
[container]          ErrorDisable
[container]          PerfMgmt
```



```
[container] RCC
[container] FrequencySynchronization
[container] HwModuleProfileConfig
[container] MPLSStatic
[container] XML
[container] Tpa
[container] MLD
[leaf] RPIsolationEnabled
[leaf] RPIsolationMultiple
[container] AMT
[container] PriorityFlowControlWatchdog
[container] SSH
[container] BNG_PBR
<snip>
```



警告：请注意，登录设备需要凭证。这些凭证是设备的本地凭证，并且需要root-Is/admin配置文件。

下一个示例显示如何使用XML API构建查询。例如，查询必须检查XML代理是否已启用：

```
xml-schema[config]:> ls
[container]          RIP
[container]          TCL
[container]          LawfulIntercept
[container]          ErrorDisable
[container]          PerfMgmt
[container]          RCC
[container]          FrequencySynchronization
[container]          HwModuleProfileConfig
[container]          MPLSStatic
[container]          XML    >>> Here
[container]          Tpa
[container]          MLD
[leaf]               RPIsolationEnabled
[leaf]               RPIsolationMultiple
[container]          AMT
[container]          PriorityFlowControlWatchdog
[container]          SSH
[container]          BNG_PBR
[container]          L2TP
[container]          Exception
[container]          IP_RAW
[container]          MSTAG
[container]          FpdXRConfig
```

```
xml-schema[config]:> cd XML
```

```
xml-schema[config]:XML> ls
[container]          Agent
```

```
xml-schema[config]:XML> cd Agent
```

```
xml-schema[config]:XML.Agent> ls
[container]          Default
[container]          SSL
[container]          TTY
```

```
xml-schema[config]:XML.Agent> cd TTY
```

```
xml-schema[config]:XML.Agent.TTY> ls
[leaf]               Enable    >>> Leaf of interest.
[leaf]               IterationSize
[leaf]               StreamingSize
[container]          Throttle
[container]          Session
```

注意下一个提示：

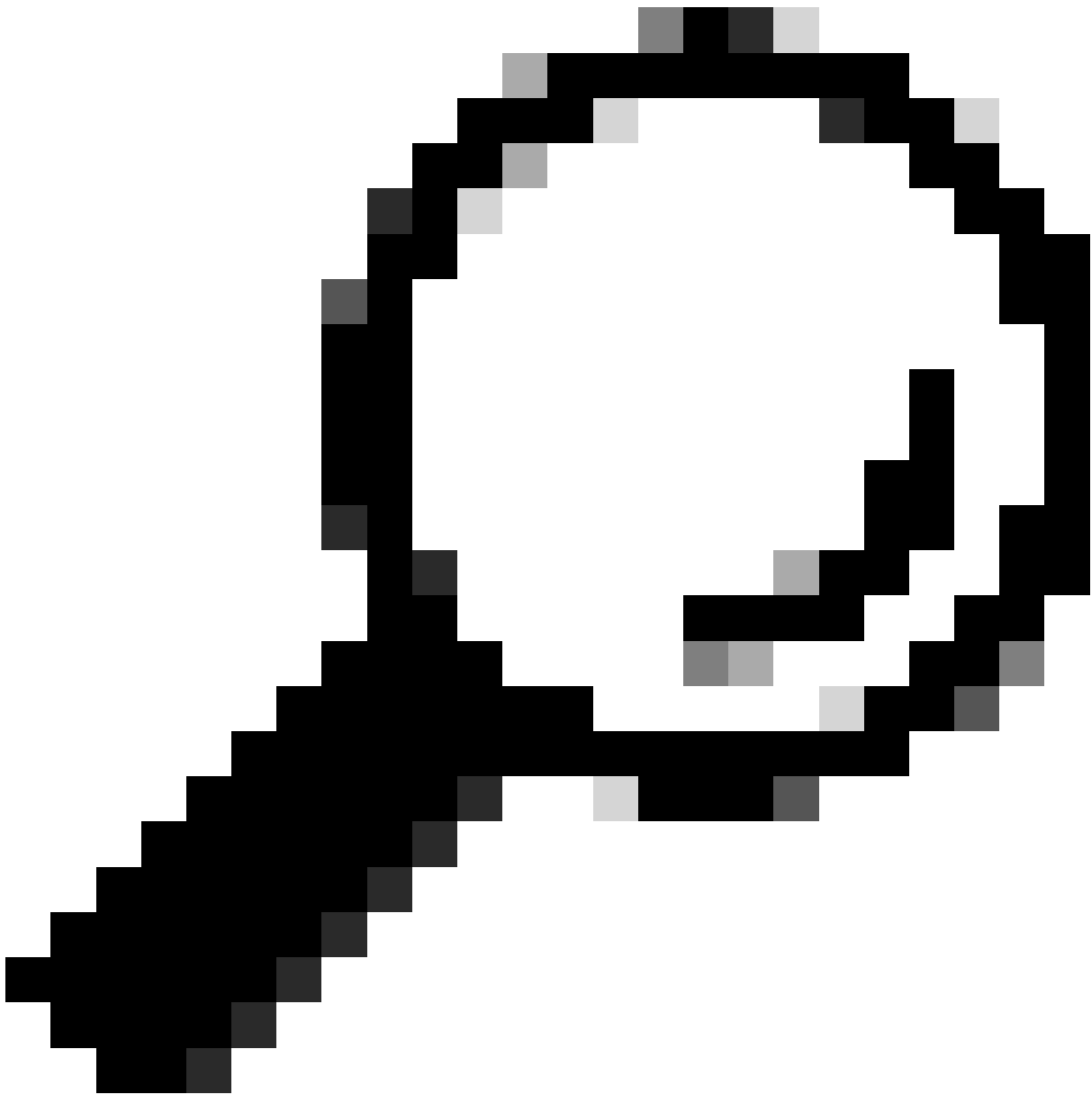
```
xml-schema[config]:XML.Agent.TTY>
```

上一个提示符显示XML树。使用此信息时，查询是：

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get>
    <Configuration>
      <XML>
        <Agent>
          <TTY></TTY>
        </Agent>
      </XML>
    </Configuration>
  </Get>
</Request>
```

测试XML查询

构建查询后，下一步是对其进行测试。要测试它，可以在CLI中发出xml echo format命令。此操作可在同一设备中存档。



提示：请注意，输入上一个命令后，用户可以复制/粘贴查询并按Enter键。设备显示对查询的响应。

```
RP/0/RP0/CPU0:XR#xml echo format
XML>
XML> <?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get>
    <Configuration>
      <XML>
        <Agent>
          <TTY></TTY>
        </Agent>
      </XML>
    </Configuration>
  </Get>
</Request>    >>> Hit enter here. Immediately getting the reply.
```

```
! Reply:
<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0">
  <Get>
    <Configuration>
      <XML MajorVersion="1" MinorVersion="4">
        <Agent>
          <TTY>
            <Enable>
              true >>> This is the requested value.
            </Enable>
          </TTY>
        </Agent>
      </XML>
    </Configuration>
  </Get>
  <ResultSummary ErrorCount="0"/>
</Response>
XML> exit
```

路由器显示响应和ResultSummary。



注意：此XML响应不显示，也不要求配置任何限制内存。由于未使用任何SSH/TTY，因此无论大小如何，系统都会请求响应。一个是XML客户端，另一个是嵌入式XML echo命令，行为可以改变。

思科IOS XR中的XML深入探讨

Cisco IOS XR是一个模块化系统，需要不同的软件包才能运行。XML被认为位于管理平面子系统中。

当XR启动时，在设备中利用XML的方法有两种：

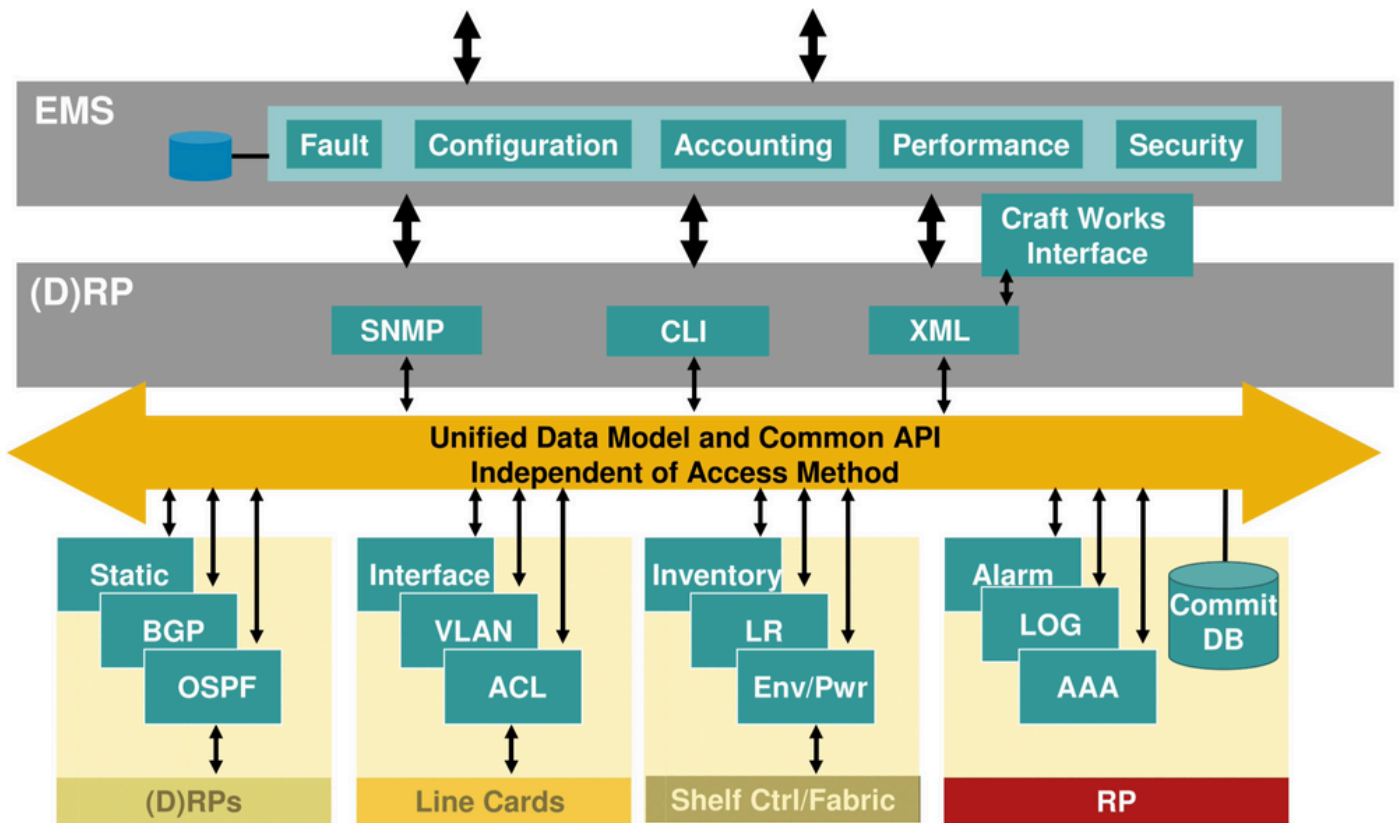
1. CORBA代理（使用SSL）。
2. SSH代理。



注意：CORBA在3.7版之后已弃用。

CORBA作为传统协议使用SSL，可在安全包中找到。另一方面，SSH方法需要使用可管理性包。

基础设施按以下方式分布：



在上图中，路由器和交换机处理器(RSP)中显示XML的主要过程。这些进程有一个通用API，用于监控来自设备中其他进程的所有信息，这些进程是：

- 从线卡：
 - 统计信息收集器
 - 统计信息服务器
- 从路由器处理器：
 - 统计信息管理器。
 - SysDB。
 - TCP(用于XML API的TCP 38751)
 - NetIO

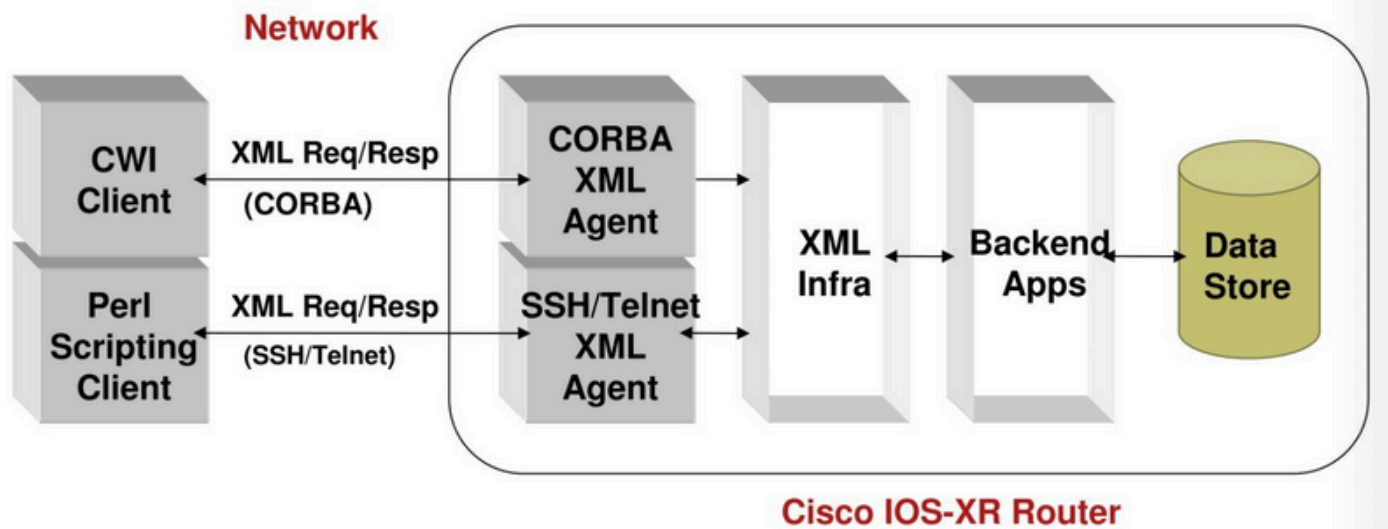


注意：要查看有关此过程的详细信息，请检查Cisco IOS XR中的XML错误部分。

根据请求，将触发一个或多个进程。在开发自动化工具时，如果响应大于块大小，则响应无法成功。如果某个进程需要很长时间才能回复，它可能会生成EDM日志、关闭或影响服务。

与客户端的API交互

下一张图显示了交互：



要在设备上配置/启用XML代理，请使用：

```
RP/0/RP0/CPU0:XR#config
RP/0/RP0/CPU0:XR(config)#xml agent tty
RP/0/RP0/CPU0:XR(config-xml-tty)#commit
```

系统需要与客户端在以下方面达成一致：

1. 版本
2. 方案

版本可在以下两个区域中进行定义：

- 请求标记：

```
<Request Major Version="1" MinorVersion="0">
```

- 主要组件标签：

以下哪一项适用于特定组件：

```
<BGP MajorVersion="1" MinorVersion="0">
```



注意：请记住，您可以检查架构版本，并将GetVersionInfo操作应用于容器。

-
- 次要版本更新：对XML架构的任何添加，如添加新数据项。
 - 主要版本更新：语义更改、对架构或组件的删除等。

您可以检查版本。下一个示例显示如何执行此操作：

注意：XML操作包括GET、SET等。

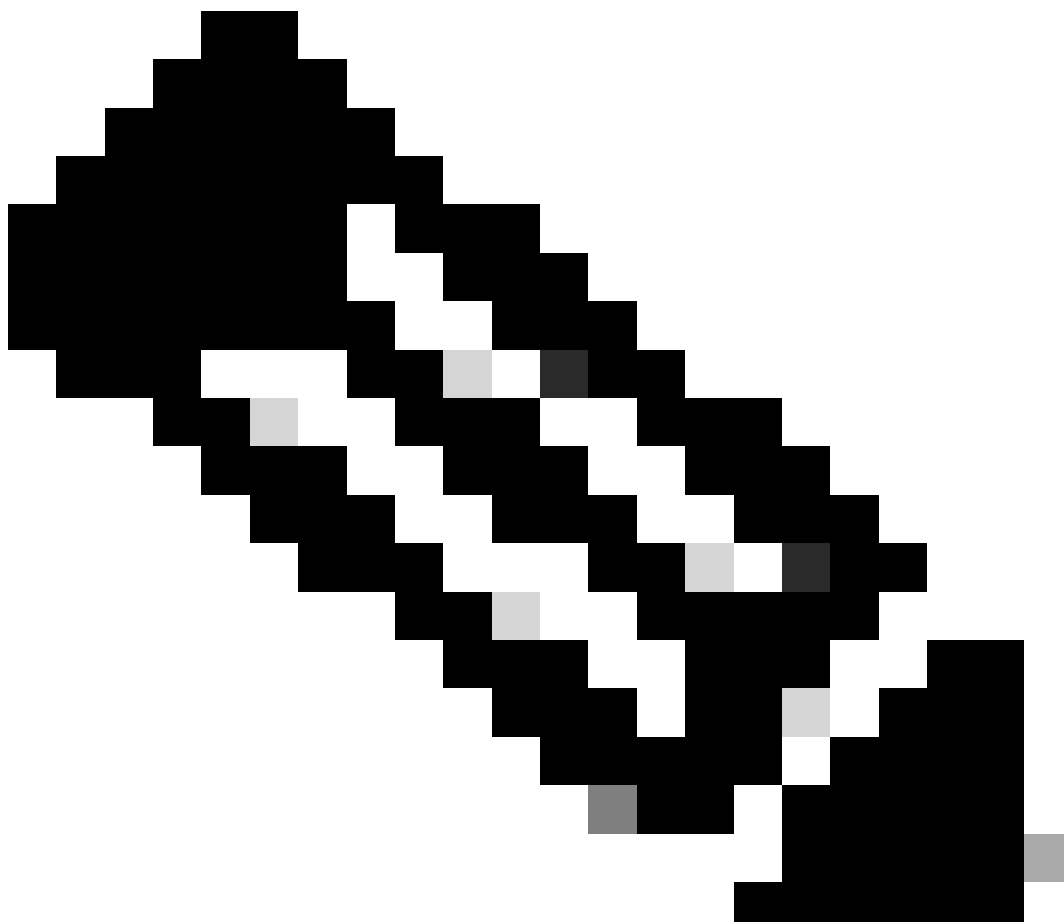
```
<?xml version="1.0" encoding="UTF-8"?>
<Request Major Version="1" MinorVersion="0">
<GetVersionInfo>
<Operation/>
</GetVersionInfo>
</Request>
```

使用xml echo format命令，然后向查询中添加标记GetVersionInfo，以查找在路由器上运行的版本。如下所示：

```
RP/0/RSP0/CPU0:XR#xml echo format
Mon Jul 31 13:53:50.993 UTC
XML> <?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
```

```
<GetVersionInfo>
  <Configuration>
    <XML>
      <Agent>
      </Agent>
    </XML>
  </Configuration>
</GetVersionInfo>
</Request> >>> This is the request
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0">
  <GetVersionInfo>
    <Configuration>
      <XML MajorVersion="1" MinorVersion="4">
        <Agent>
          <Default>
            <VRFTable>
              <VRF>
                <ApplyGroup MajorVersion="2" MinorVersion="3"/>
                <ExcludeGroup MajorVersion="2" MinorVersion="3"/>
                <ApplyGroupAppend MajorVersion="2" MinorVersion="3"/>
                <ApplyGroupRemove MajorVersion="2" MinorVersion="3"/>
              </VRF>
            </VRFTable>
          </Default>
          <SSL>
            <VRFTable>
              <VRF>
                <ApplyGroup MajorVersion="2" MinorVersion="3"/>
                <ExcludeGroup MajorVersion="2" MinorVersion="3"/>
                <ApplyGroupAppend MajorVersion="2" MinorVersion="3"/>
                <ApplyGroupRemove MajorVersion="2" MinorVersion="3"/>
              </VRF>
            </VRFTable>
          </SSL>
        </Agent>
      </XML>
    </Configuration>
  </GetVersionInfo>
  <ResultSummary ErrorCount="0"/>
</Response>
XML>
```



注：请求显示主要组件中所有正在运行的版本，并在组件中的容器中显示版本。

Cisco IOS XR中的XML错误

如果路径正确，每个XML API请求都会显示请求的信息。

查询出错时

路由器会显示三种不同的消息：

- ItemNotFound

每当GET操作具有空响应时，都会显示此消息。

- ItemNotFoundBelow

GET操作在XML架构中不包含此操作。

- 未找到

请求的元素找不到元素级别。

错误类型

1. 传输：此类错误包括XML代理/客户端通信之间的任何错误。这意味着传输中可能发生任何SSH交互或问题。因此，为了检查此类问题，建议检查SSH踪迹，以检查任何身份验证、端口等问题。

2. XML解析器：格式和语法的任何问题，发送的响应或查询中的问题。当发生错误时，这些问题通常会发送故障原因。

例如：

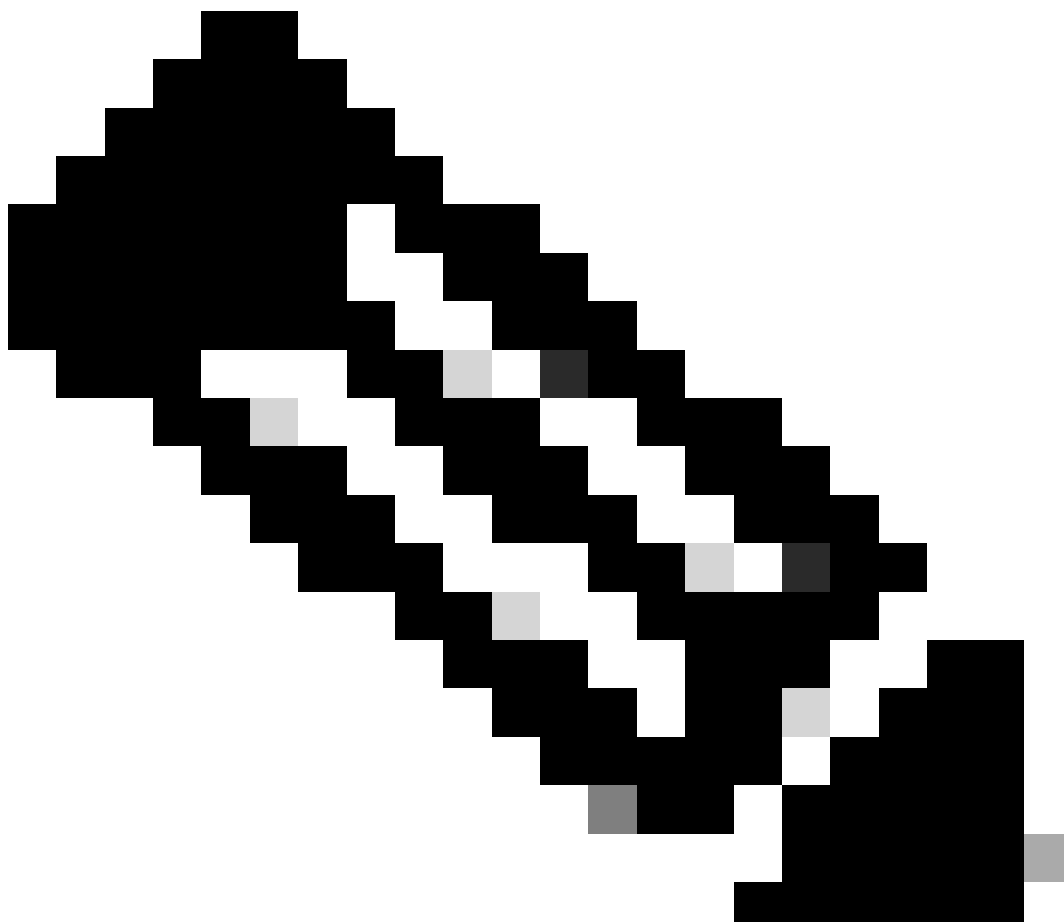
```
ERROR: 0xa367a600 'XML Service Library' detected the 'fatal' condition 'The throttle on the memory usage'
```

3. XML方案：与路由器方案不同的任何方案。要解决这些问题，请检查CLI方案和版本。

例如：

```
ErrorCode="0x43688400" ErrorMessage="'XMLMDA' detected the 'warning' condition 'An'
```

4. 操作处理：配置设备时，可能会出现这些问题。要解决此问题，您需要对进程（例如commit、sysdb等）进行故障排除。



注：错误信息会添加到工序要素层。这以ErrorCode（32位int）和ErrorMsg属性的形式编码。

其他建议

其他有用技术：

1. 通配符：也称为特定查询。
2. 批处理：将多种技术或操作合并到一个请求中（尽力操作）。
3. 自定义筛选：如果架构允许，可帮助选择表中的行。

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。