

Fehlerbehebung bei XML für Cisco IOS XR

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Drosselung von Speicherproblemen](#)

[Weitere Problembeschreibung](#)

[Lösung](#)

[XML in Cisco IOS XR](#)

[Verwenden der XML-API](#)

[Von XML-API unterstützte Vorgänge](#)

[Angebotene XML-Schemadienste](#)

[Erstellen einer Abfrage](#)

[Testen von XML-Abfragen](#)

[XML - Vertiefung in Cisco IOS XR](#)

[API-Interaktion mit Client](#)

[XML-Fehler in Cisco IOS XR](#)

[Wenn die Abfrage falsch ist](#)

[Fehlertypen](#)

[Weitere Empfehlungen](#)

Einleitung

In diesem Dokument werden Probleme mit der Drosselung des XML-Arbeitsspeichers beschrieben, die bei Serviceanfragen und Cisco Tools aufgetreten sind.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Cisco IOS® XR-Plattform
- XML-Infrastruktur (Extensible Markup Language)
- Common Object Request Broker-Architektur (CORBA)
- Element Management System (EMS)
- Externer Datenmanager (EDM)

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf ASR9000.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Drosselung von Speicherproblemen

Es wurde ein Fall mit folgenden Fragen eröffnet:

1. Wie viel Speicher kann maximal auf dem Router konfiguriert werden?
2. Gibt es eine Möglichkeit, große XML-Antworten aufzubrechen?

Antwort:

1. Die maximale Speicheränderung hängt von der Version ab, die der RSP/Knoten ausführt (Versionen für cXR und eXR)

In 64-Bit-Version (eXR). Sie haben eine Drosselspeichergröße:

```
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)
```

In 32-Bit-Version (cXR). Sie haben ein anderes Limit:

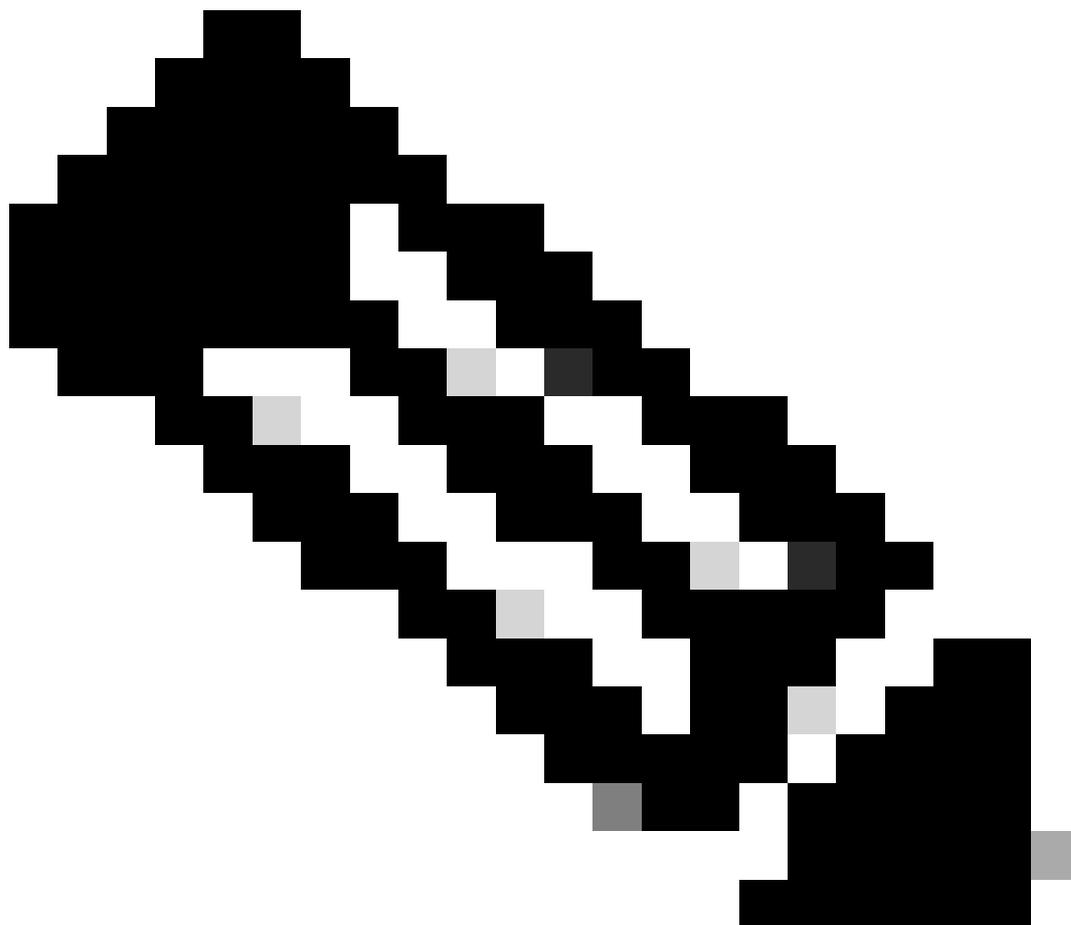
```
RP/0/RSP0/CPU0:XR#show version
Cisco IOS XR Software, Version 6.4.2[Default]
```

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)
```



Hinweis: Der Standardwert in beiden Versionen beträgt 300 MB.

2. Es gibt verschiedene Möglichkeiten, die XML-Antwort zu unterbrechen:

- Iteratoren:

Wenn Iteratoren konfiguriert sind, segmentieren Sie die XML-Gesamtreaktion in einer

bestimmten Fenstergröße. Das Fenster hat die Größe des Iterators. Wenn die Antwort beispielsweise 1 GB und die Iteratorgröße 500 MB beträgt, segmentieren Sie die XML in 2 Antworten.

Diese Methode ändert die Antwort, indem sie (basierend auf der Iteratorgröße) GetNext-Vorgänge hinzufügt (was die Antwort benötigt).

Eine Sitzung kann bis zu 10 Iteratoren umfassen.

- Drosselung (Prozessgeschwindigkeit):

Diese Funktion schränkt den vom XML-Prozess belegten Arbeitsspeicher ein. Wenn ein Prozess den Speicher überläuft, antwortet er mit der Fehlermeldung: "Die Drosselung der Speichernutzung wurde erreicht".

- Streaming:

Ähnlich wie bei Iteratoren segmentiert Streaming die XML-Antwort in bestimmte Fenster. Der Unterschied besteht in der Antwort. Der GetNext-Vorgang und die Iterator-ID werden entfernt. Die XML sendet die gestreamten Pakete, und der Client erstellt die Antwort, wenn der Stream beendet wird.

Weitere Problembeschreibung

Für Anwendungsfälle, bei denen eine Automatisierung erforderlich ist, verwenden Sie das Tool [pyIOSXR](#). Dieses Automatisierungstool ist ein XML-Agent, es hilft, einige show-Befehle auszugeben, und stellt insgesamt eine Verbindung zum Gerät her.

Jedes Mal, wenn Sie eine große Anfrage an diesen Agenten senden, wird ein Fehler angezeigt:



Hinweis: Nur registrierte Cisco BenutzerInnen können auf interne Cisco Tools und Informationen zugreifen.

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

Wenn die Fehlermeldung "Optimize the request to query small data" angezeigt wird, können Sie die XML-API des Agenten mit Iteratoren konfigurieren. Theoretisch ermöglicht dies eine Segmentierung der Reaktion.

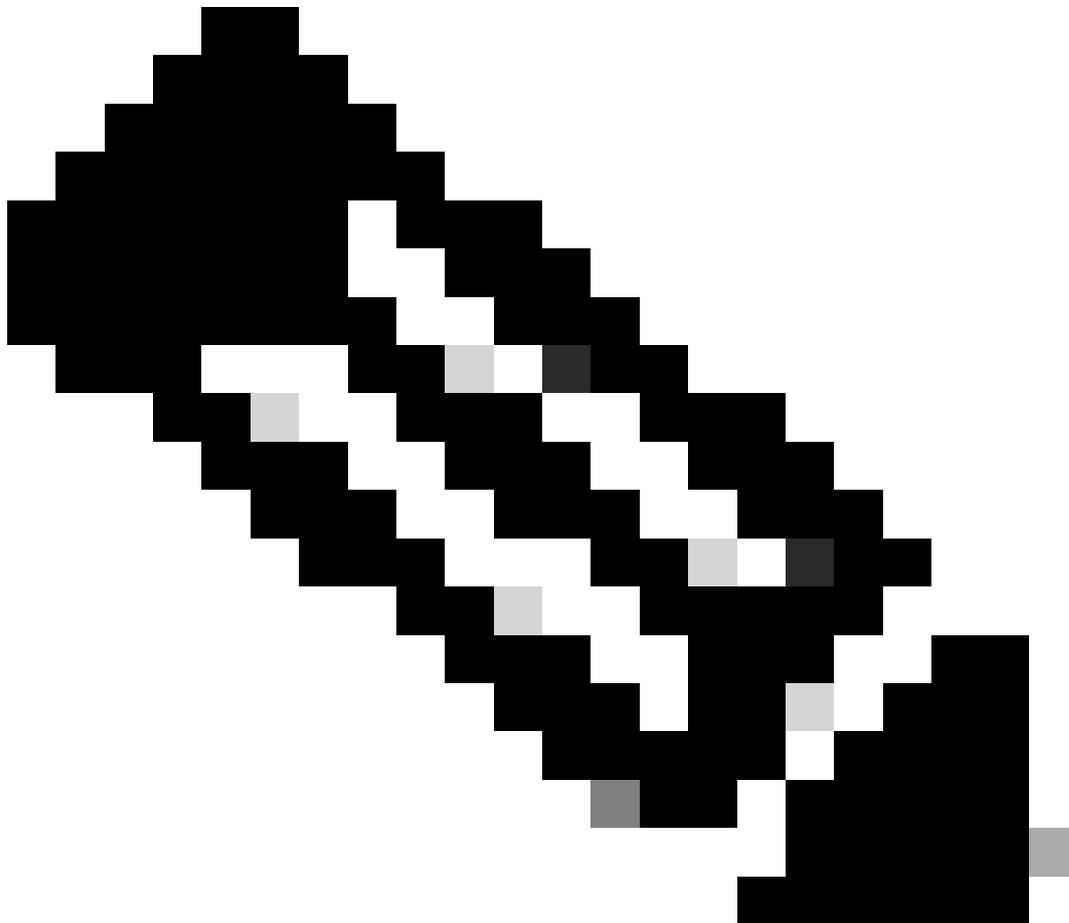
Bei der Konfiguration von Iteratoren wurde dieselbe Fehlermeldung angezeigt:
"ANTWORTFEHLER ... Bitte optimieren Sie die Anfrage, um kleinere Daten abzufragen."

Wenn ein Fehler angezeigt wird, ist der nächste Schritt zu verstehen, warum die Funktion, in diesem Fall Iteratoren, nicht für die Abfrage funktioniert hat.

Der [pyIOSXR](#) empfiehlt die korrekte Aktivierung des XML-Agenten im Gerät, was bedeutet, dass

die API die Verwendung von Iteratoren nicht zulässt.

Der nächste Schritt ist, die zweite Option zu testen: Streaming.



Hinweis: pyIOSXR erlaubt es uns nicht, mit anderen Headern als der XML-Abfrage zu arbeiten, die die folgenden Elemente haben:

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

Streaming und Iteratoren fügen den Abfragen einen Header hinzu. Streaming fügt eine Stream-ID

hinzu, die dem System dabei hilft, die vollständige Antwort zu erstellen. Iteratoren fügen GET_Next und Iterator ID hinzu.

Beim Testen des Streamings wurde die gleiche Fehlermeldung wie bei Iteratoren angezeigt.

Lösung

Im Abschnitt "More Recommendation to do Queries to the Device" (Weitere Empfehlungen für Abfragen an das Gerät) finden Sie weitere Tools, um diese Probleme zu beheben. Platzhalter sind einer davon. Ein Platzhalter ist die Lösung für die Speicherdrosselungsgrenze.

Platzhalter erstellt eine spezifische Abfrage, um unnötige Informationen nicht anzufordern. Für BGP-Informationen verwenden Sie beispielsweise den Befehl show route bgp anstelle des generischen Befehls show route. Dieses Beispiel gilt für die XML-Abfragen und die XML-Logik. Das Anfordern von Masseninformationen an das System kann zu Speicher- und Verarbeitungsproblemen führen.

Beim Öffnen des Gehäuses wurde diese Abfrage verwendet:

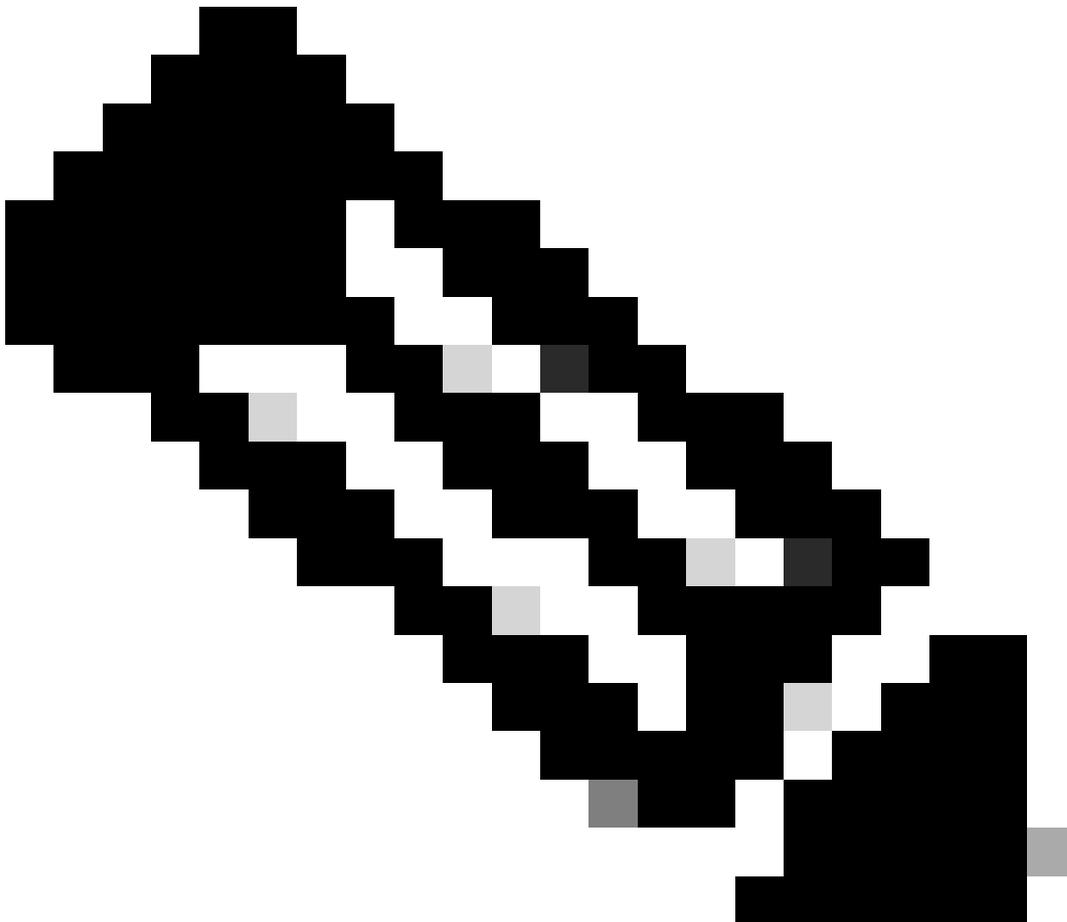
```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get>
    <Operational>
      <BGP MajorVersion="36" MinorVersion="1">
        <InstanceTable>
          <Instance>
            <Naming>
              <InstanceName>
                DEFAULT
              </InstanceName>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

```
</Instance>
  </InstanceTable>
</BGP>>
</Operational>
</Get>
</Request>
```

Diese Abfrage bezieht sich auf die vollständigen BGP-Tabellen. Für diese unspezifische Anforderung betrug die Antwort fast 2,2 Gb, daher wird die Speicherdrosselgrenze erreicht.

Um dies zu beheben, ist eine bestimmte Abfrage erforderlich. Dadurch kann das System die Abfrage verarbeiten und die Informationen zurückgeben.

XML in Cisco IOS XR



Hinweis: Falls Sie weitere Informationen, Spezifikationen oder andere Befehle benötigen, klicken Sie auf den nächsten Link XML Programming guide: [XR XML Programming Guide](#)

Der XML-Code definiert, wie Daten angezeigt und strukturiert werden. Auf diese Weise können Sie analysieren, was der Computer als Bits versteht, und strukturierte, standardisierte Informationen anzeigen.

XML hat diese Struktur:

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

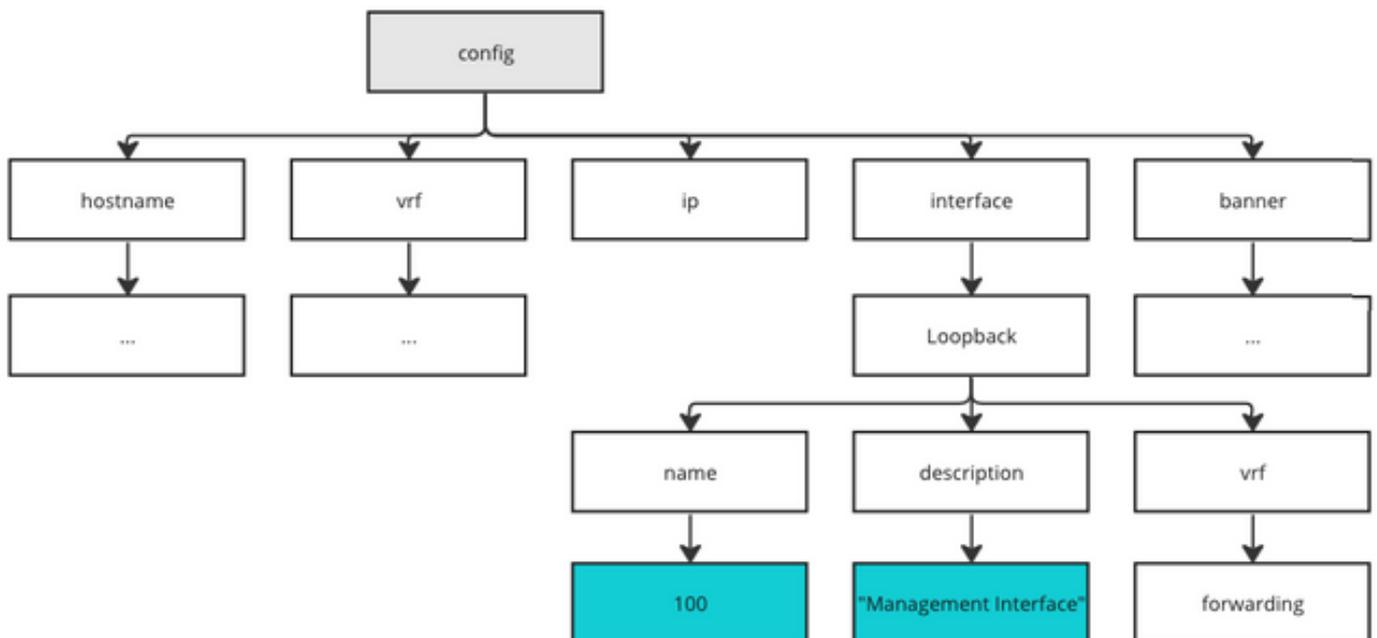
Alle Tags bestehen aus zwei Teilen, dem öffnenden Tag (<init>) und dem schließenden Tag (</init>). Wenn diese Struktur nicht verwendet wird, kann XML nicht nachvollziehen, wo ein Tag endet.

XML besteht aus zwei Einheiten:

- Container
- Blätter

Eine Analogie für diese strukturierten Daten ist der Vergleich von XML-Daten als Baum. Ein Container ist ein Ast und jeder Ast hat stubbed Blätter. Leafs enthalten keine anderen als Informationen.

Der nächste Baum hat z. B. die Wurzel in Grau, Behälter in Weiß und Blätter in Blau.

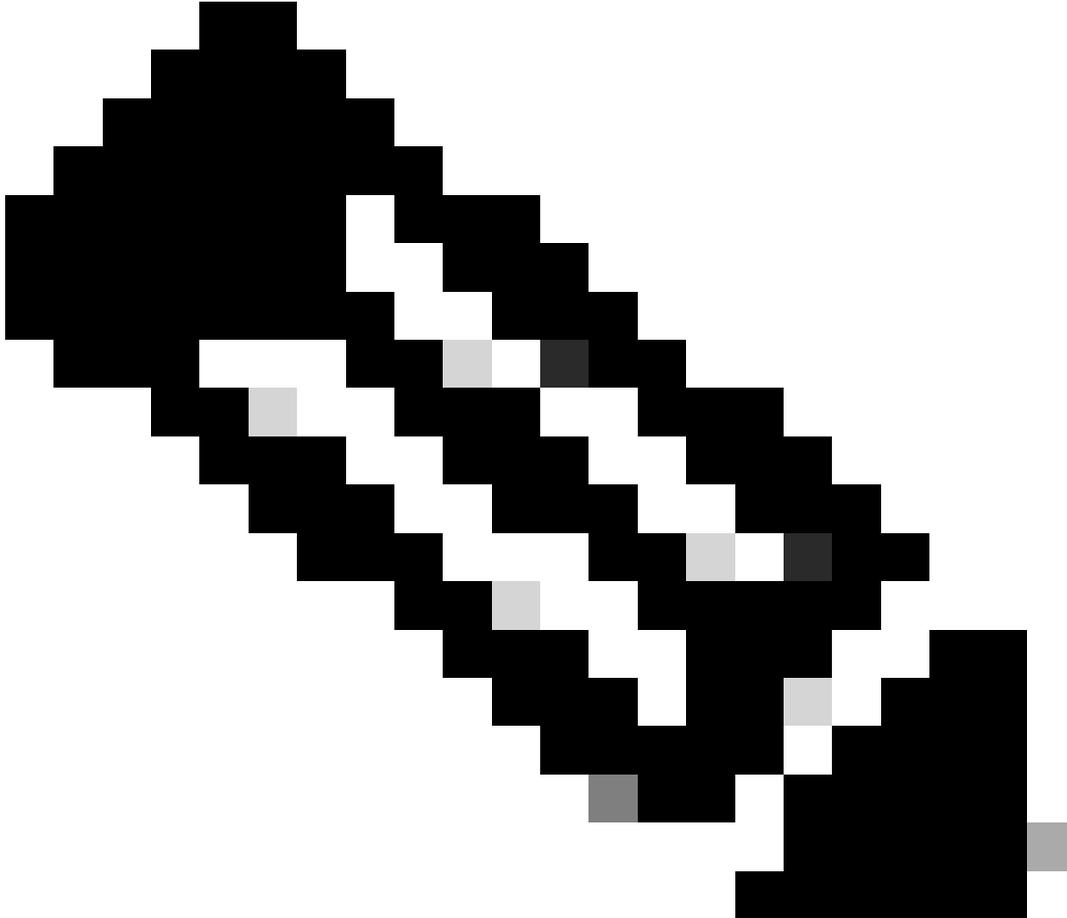


Verwenden der XML-API

Um die XML-API zu testen bzw. zu nutzen, benötigen Sie als Erstes eine Abfrage.

1. Die Abfrage hat einen Header:

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



Hinweis: Dies ist das einzige Tag, das kein schließendes Tag benötigt.

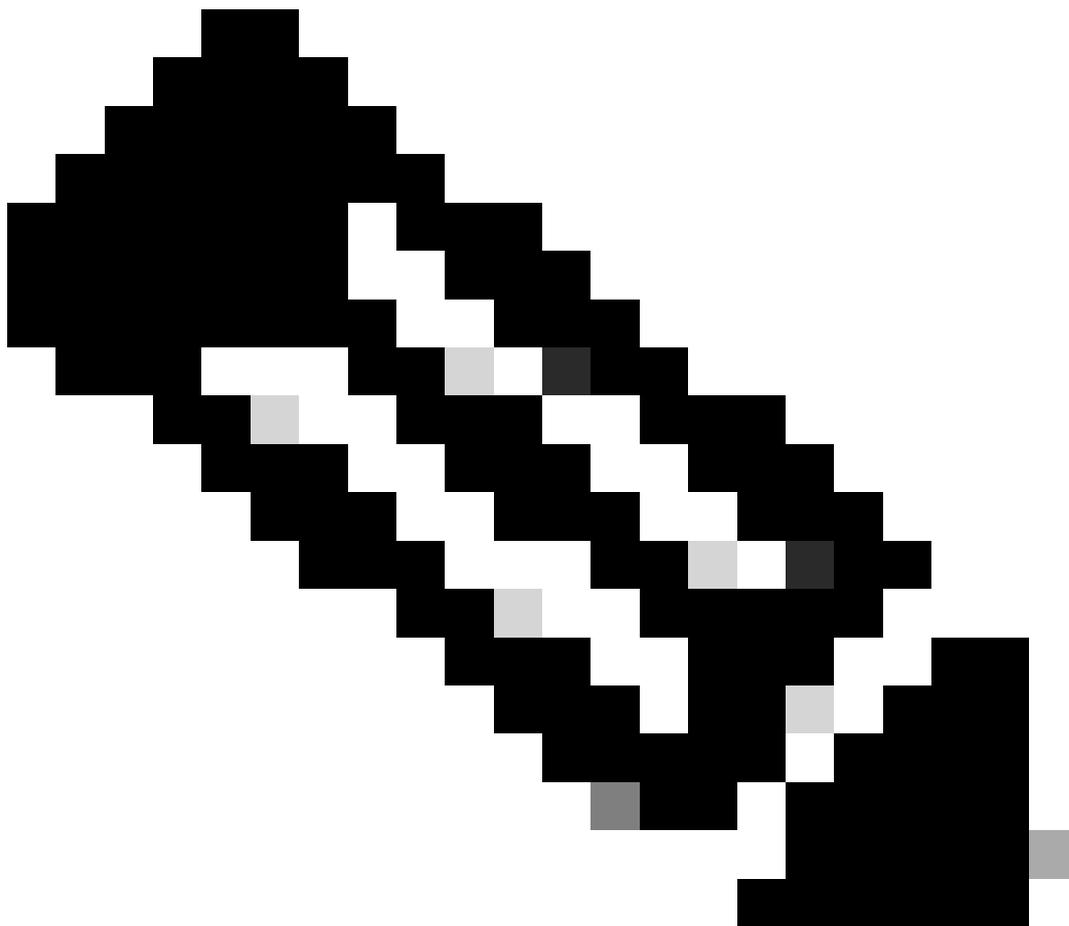
2. Sie müssen einen Tag für die Anforderung hinzufügen. In der Anforderung müssen Sie die Version angeben.

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

3. Mit dem Header, der Anforderung und der Version wird der Text mit jeder XML-Operation

fortgesetzt, über die die API verfügt:

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



Hinweis: Beachten Sie, dass Anforderungen unterschiedliche Vorgänge in jeder Abfrage enthalten. Es ist nicht erforderlich, pro Vorgang eine Anforderung auszuführen.

Im nächsten Beispiel wird eine Anforderung mit allen erforderlichen Informationen angezeigt:

```
<?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>
```

Von XML-API unterstützte Vorgänge

Cisco IOS XR unterstützt fünf Vorgänge, mit denen Benutzer mit den Informationen des XML-Schemas interagieren können:

1. Systemeigene Datenvorgänge:

- `<Get></Get>`: Sammeln von Aktionsdatenelementen (Daten, die Leaf enthält).
- `<Set></Set>`: So konfigurieren Sie Datenelemente, ändern sie oder fügen sie hinzu.
- `<Löschen></Löschen>`: Beseitigen Sie ein oder mehrere Datenelemente.
- `<GetVersionInfo></GetVersionInfo>`: Vorgang zum Abrufen der Haupt- und Nebenversionsnummern der angeforderten Komponenten.
- `<GetDataSpaceInfo></GetDataSpaceInfo>`: Diese operativen Tags zeigen die Blattnamen an, die den Containern zugeordnet sind.

2. CLI-Vorgänge:

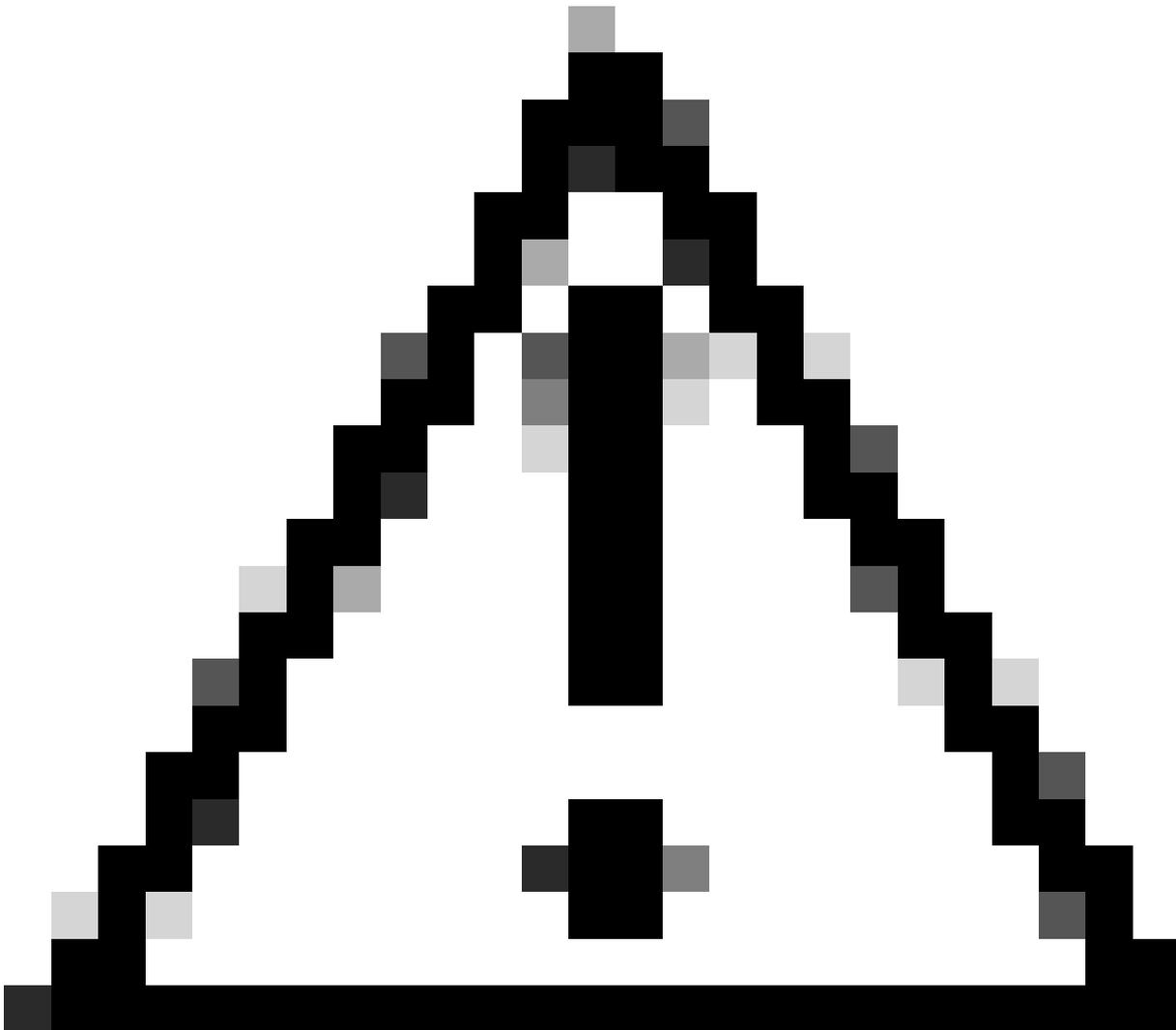
- `<CLI></CLI>`: Tag, mit dem eine Konfigurationsanforderung als CLI-Befehl ausgegeben werden kann.
- `<GetNext></GetNext/>`: Wenn die Daten größer als die Blockgröße sind, wird dieses Tag zur nächsten segmentierten Antwort hinzugefügt.

Angebotene XML-Schemadienste

Die XML-API implementiert die folgenden Namespaces:



Hinweis: Falls Sie Root-Objekte sehen müssen, besuchen Sie: [Cisco IOS-XR XML Network Management](#).



Achtung: Beachten Sie, dass jeder Namespace unterschiedliche Daten und Vorgänge überwacht.

-
- Konfiguration: Vorgang, der die Aktionen Abrufen, Festlegen und Löschen ermöglicht.
 - Operativ: Wie CLI-Befehle anzeigen.
 - Aktion: Unterstützen Sie die Aktion Festlegen. Mit dieser Operation können Sie auf Aktionsdaten zugreifen, wie z.B. Clear-Befehle und andere ähnliche Befehle.
 - AdminOperational (Admin-Betrieb): Zugriff auf Admin-Betriebsdaten.
 - AdminAction (Admin-Aktion): Zugriff auf die Verwaltungsaktionsdaten
 - Admin Configuration: Zugriff auf die Administrationskonfigurationsdaten.

Erstellen einer Abfrage

Es gibt zwei verschiedene Wege zu reisen. Nachdem Sie die Namespaces für die Abfrage ausgewählt haben, müssen Sie ein Schema verwenden, mit dem Sie arbeiten können:

1. XML-Schemadokumentation

Diese Optionen bieten eine vollständige XML-Struktur, die Container und Leafs anzeigt. Klicken Sie für die Dokumentation auf [CRS XML Schemas](#).

Die Dokumentationsseite wird angezeigt.

CISCO
Products Support Buy Training & Events Partners

Cisco Carrier Routing System

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

Auf dieser Seite werden Objekte, Container und untergeordnete Elemente angezeigt. Jedes Objekt enthält ein untergeordnetes Objekt. Wenn das untergeordnete Objekt keinen anderen Container enthält, wird es als Leaf betrachtet.

Sie können auf das untergeordnete Element klicken, und die Informationen zum Erstellen einer Abfrage werden angezeigt. Beispiel für 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>](#)

Im vorherigen Image kann für jede IPv4-Abfrage der Konfigurationsvorgang ausgeführt werden, der auf die Container-IP abzielt, im Container Cinetd, im Container Services und schließlich im Leaf IPv4.



Hinweis: Jede Operation, die ein Leaf sein muss, um die Abfrage auszuführen.

Die Abfrage wäre die nächste:

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

2. XML-API

Die XML-API ist in Cisco IOS XRI integriert. Um die XML-Struktur anzuzeigen, verwenden Sie den Befehl `show xml schema`. Durch Ausgabe des Befehls wird das gesamte XML-Schema in einer CLI ähnlich Linux angezeigt.

In dieser CLI gibt es die nächsten Aktionen:

?: Genau wie - Hilfe

LS: Auflistung der Container/Leafs im Zweigstellenbenutzer ist.

<Operation>: Mit ? können Sie überprüfen, ob die Operationen und Befehle ausgegeben werden können:

Einige Vorgänge sind:

-konfig.

- adminconfig

- CD

- Liste

- Bediener

-aktion

CD: Change Directory/container

Das Beispiel zeigt den nächsten:

```
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>
```



Warnung: Beachten Sie, dass Anmeldeinformationen erforderlich sind, um sich beim Gerät anzumelden. Diese Anmeldedaten befinden sich lokal auf dem Gerät und erfordern root-Is/admin-Profile.

Im nächsten Beispiel wird veranschaulicht, wie eine Abfrage mithilfe der XML-API erstellt wird. Beispielsweise muss die Abfrage überprüfen, ob der XML-Agent aktiviert ist:

```
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
```

Beachten Sie die nächste Eingabeaufforderung:

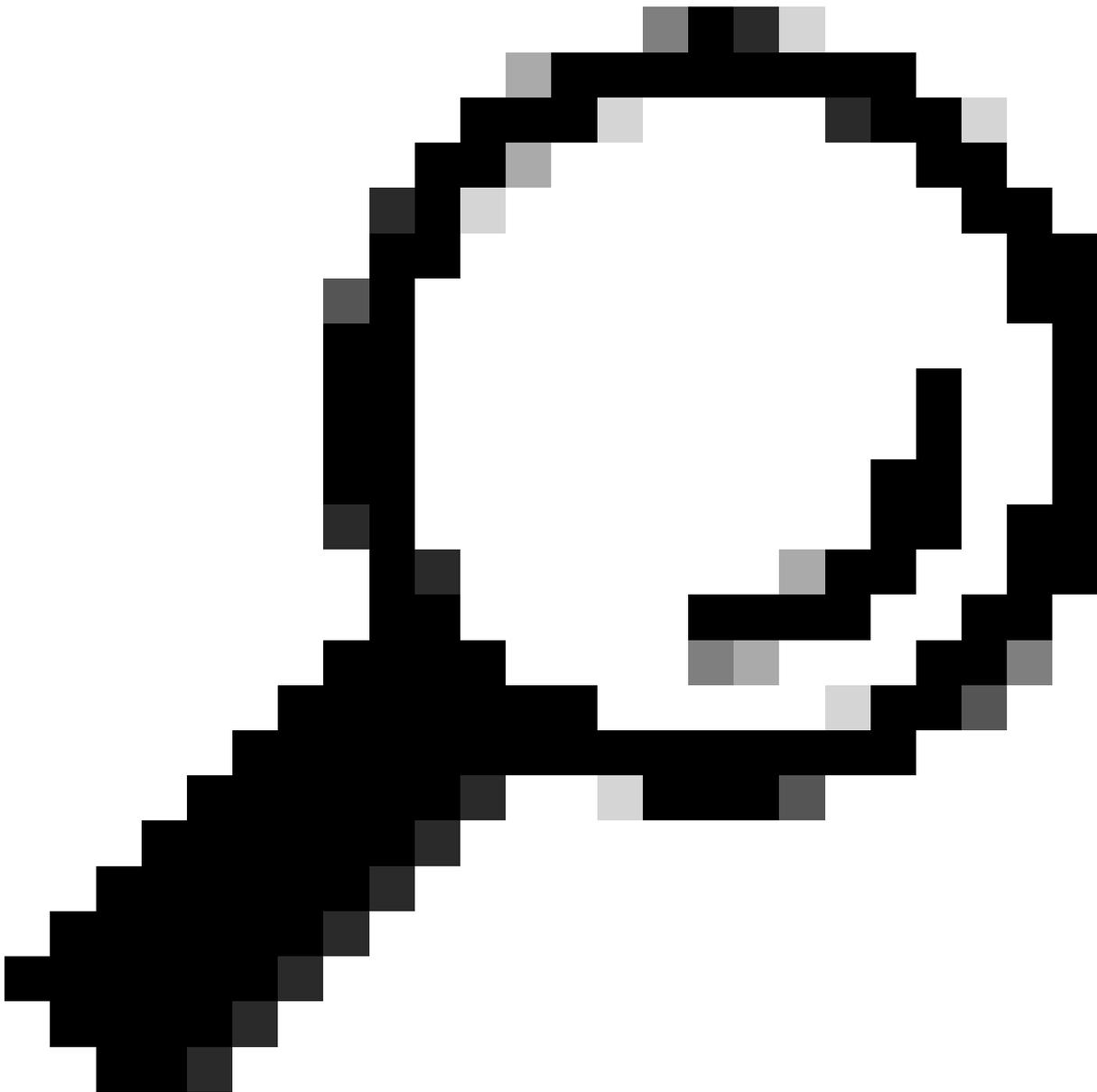
```
xml-schema[config]:XML.Agent.TTY>
```

Die vorherige Eingabeaufforderung zeigt die XML-Struktur an. Mit diesen Informationen lautet die Abfrage:

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

Testen von XML-Abfragen

Nachdem die Abfrage erstellt wurde, besteht der nächste Schritt darin, sie zu testen. Um dies zu testen, können Sie den Befehl `xml echo format` in der CLI eingeben. Diese Aktion kann auf demselben Gerät archiviert werden.



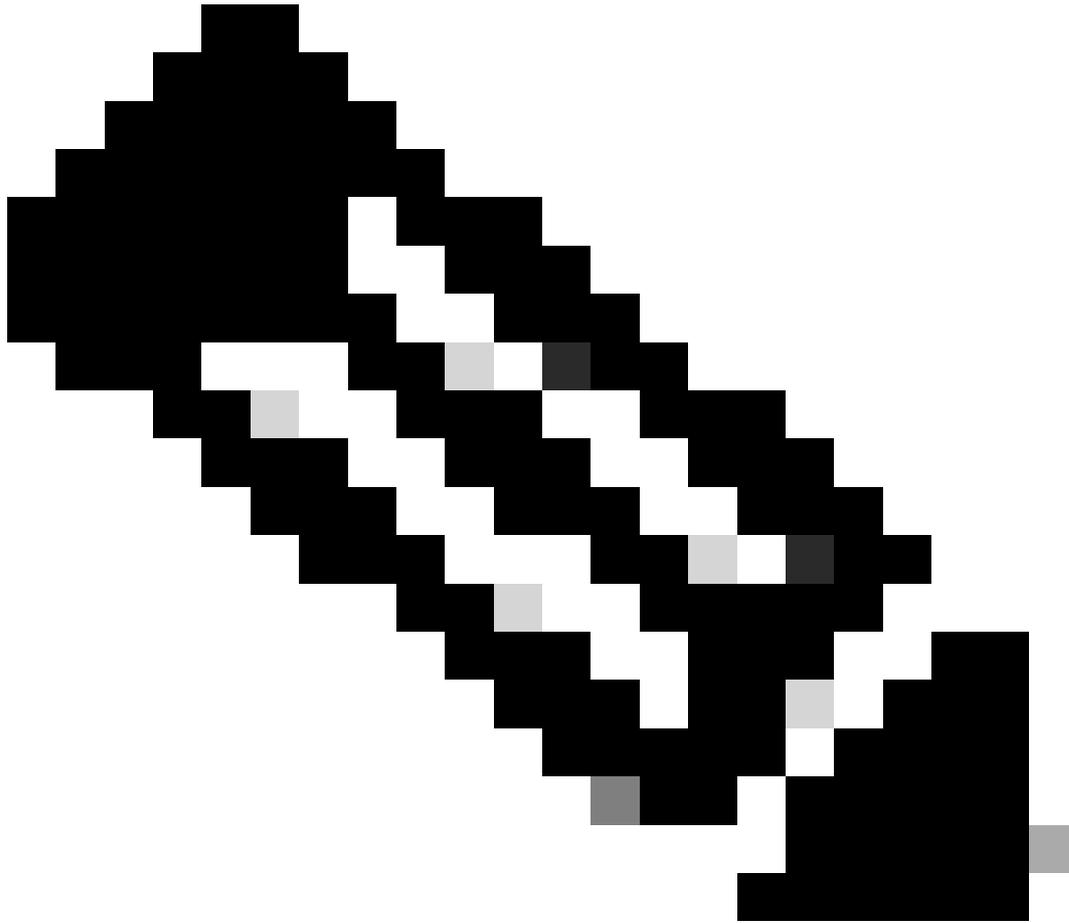
Tipp: Beachten Sie, dass Benutzer nach der Eingabe des vorherigen Befehls die Abfrage kopieren und einfügen können, indem sie die Eingabetaste drücken. Das Gerät zeigt die Antwort auf die Abfrage an.

```
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
```

Der Router zeigt Antwort und Ergebnisübersicht an.



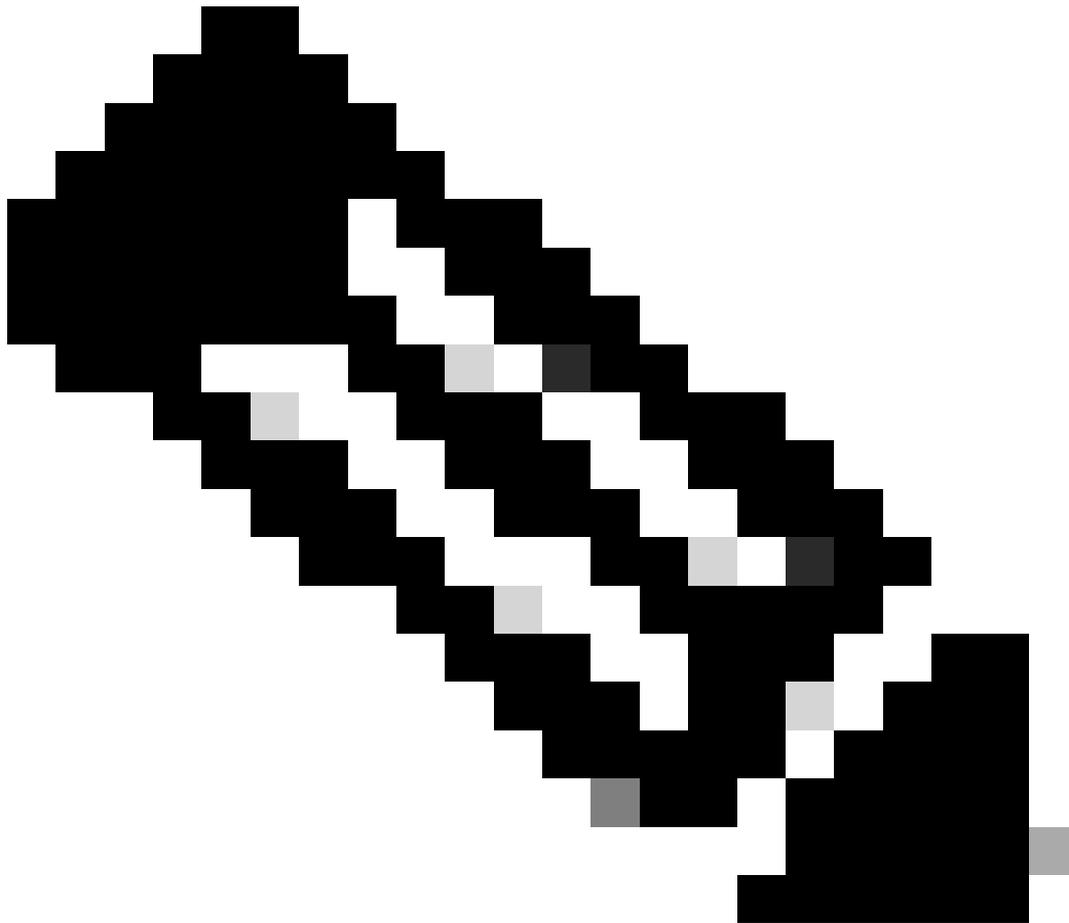
Hinweis: Diese XML-Antwort wird nicht angezeigt, und es muss kein Drosselungsspeicher konfiguriert werden. Die Antwort wird unabhängig von der Größe angefordert, da kein SSH/TTY verwendet wird. Eine Sache ist der XML-Client und andere der eingebettete XML-Echo-Befehl, das Verhalten kann sich ändern.

XML - Vertiefung in Cisco IOS XR

Cisco IOS XR ist ein modulares System, das verschiedene Pakete erfordert, um zu funktionieren. XML gilt als Teil der Managementebenenuntersysteme.

Als XR gestartet wurde, gab es zwei verschiedene Möglichkeiten, XML im Gerät zu verwenden:

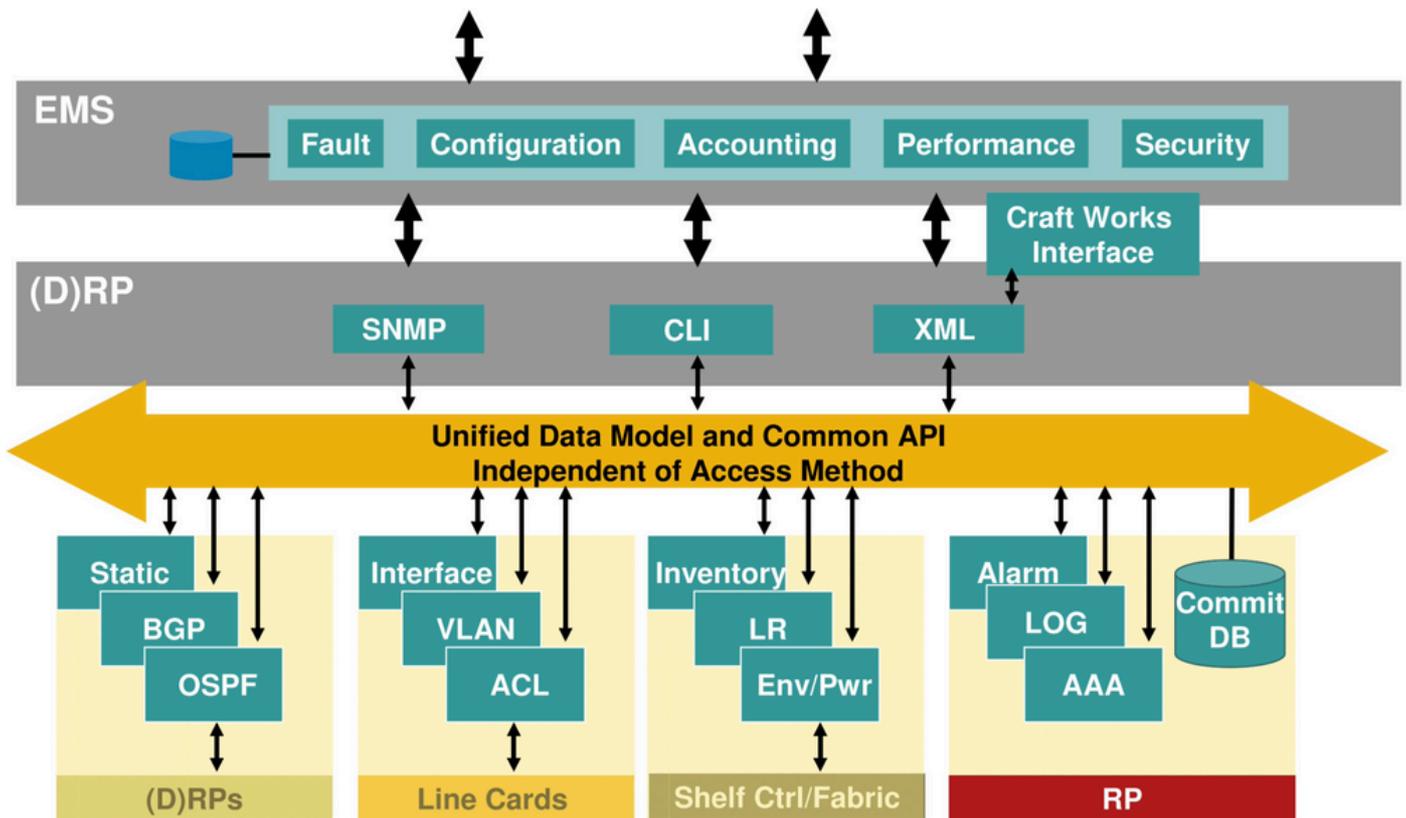
1. CORBA Agent (unter Verwendung von SSL).
2. SSH-Agent.



Hinweis: CORBA ist nach Version 3.7 veraltet.

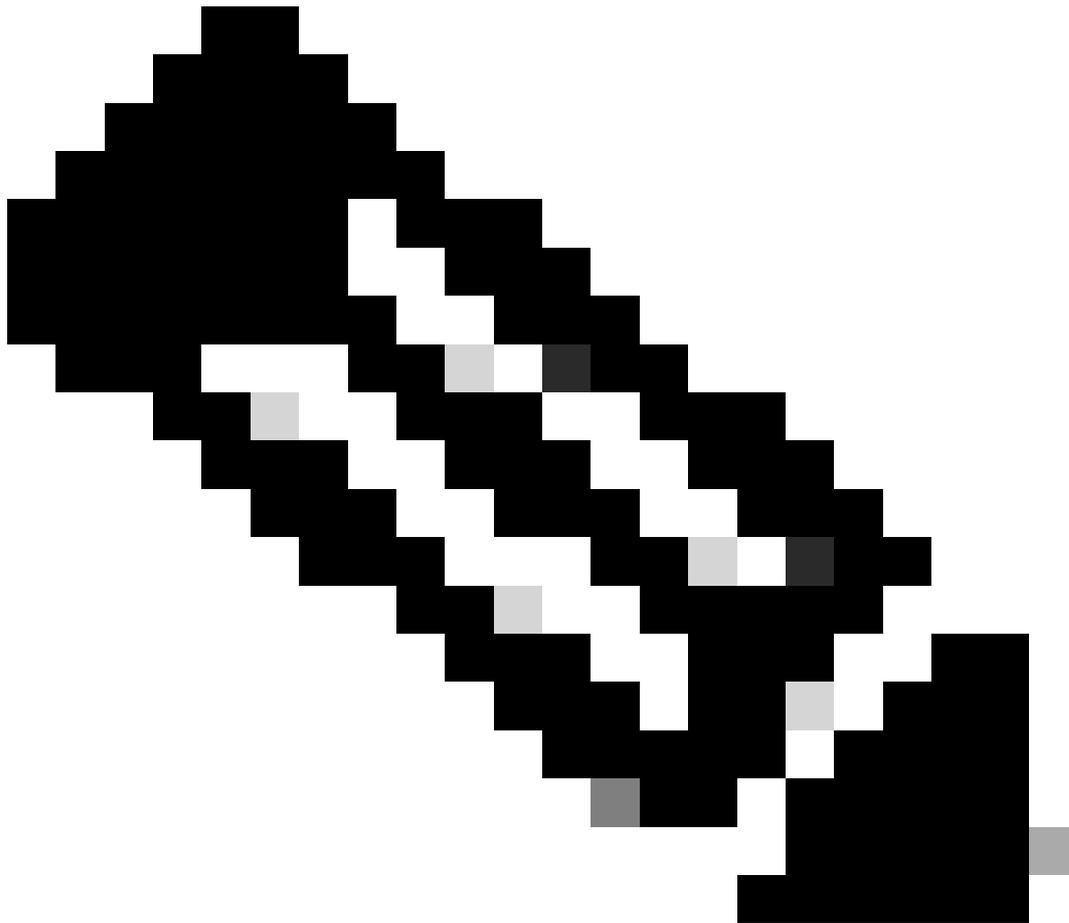
CORBA als Legacy-Protokoll verwendet SSL, die in den Security Packages zu finden ist. Andererseits muss für die SSH-Methode das Verwaltbarkeitspaket verwendet werden.

Die Infra wird auf diese Weise verteilt:



Im vorherigen Bild wird der Hauptprozess von XML im Router and Switch Processor (RSP) dargestellt. Diese Prozesse verfügen über eine gemeinsame API, die alle Informationen aus den anderen Prozessen im Gerät überwacht. Diese sind:

- Über die Linecard:
 - Statistiken-Collector
 - Stats-Server
- Vom Router-Prozessor:
 - Statistik-Manager.
 - SysDB
 - TCP (für XML-API wird TCP 38751 verwendet)
 - NetIO

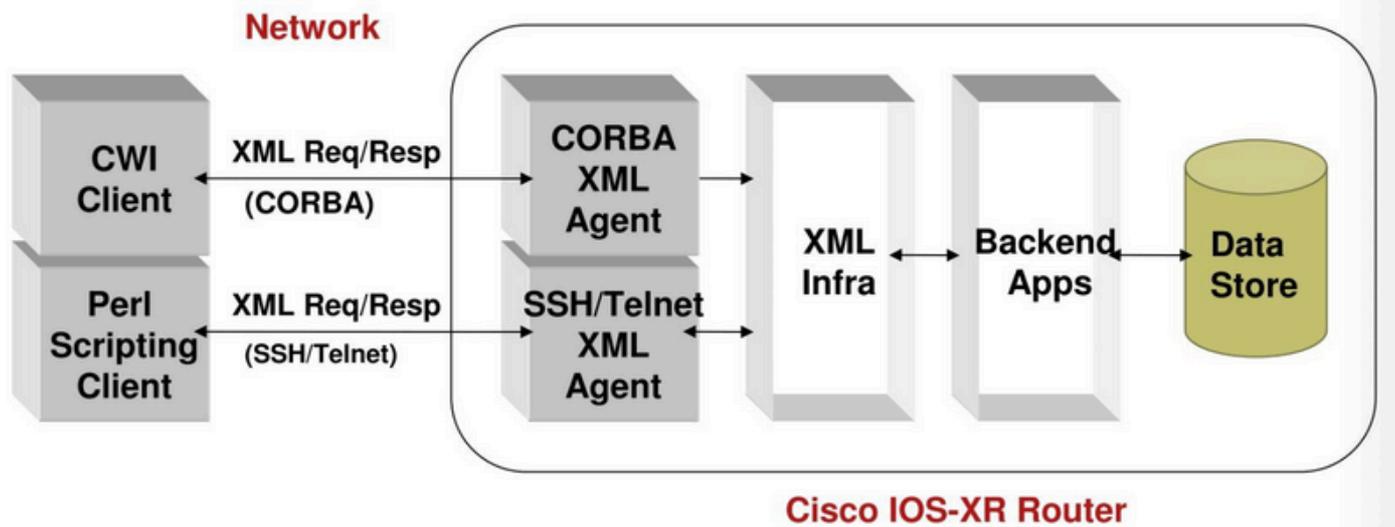


Hinweis: Weitere Informationen hierzu finden Sie im Abschnitt zu XML-Fehlern in Cisco IOS XR.

Je nach Anforderung werden ein oder mehrere Prozesse ausgelöst. Wenn die Antwort bei der Entwicklung von Automatisierungstools größer als die Blockgröße ist, kann sie nicht erfolgreich sein. Wenn die Beantwortung eines Prozesses sehr lange dauert, kann er EDM-Protokolle generieren, das System herunterfahren oder Dienste beeinträchtigen.

API-Interaktion mit Client

Das nächste Bild zeigt die Interaktion:



Verwenden Sie zum Konfigurieren/Aktivieren des XML-Agenten auf dem Gerät Folgendes:

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

Das System muss mit dem Kunden in folgenden Bereichen übereinstimmen:

1. Versionen
2. Schema

Versionen können in zwei Bereichen definiert werden:

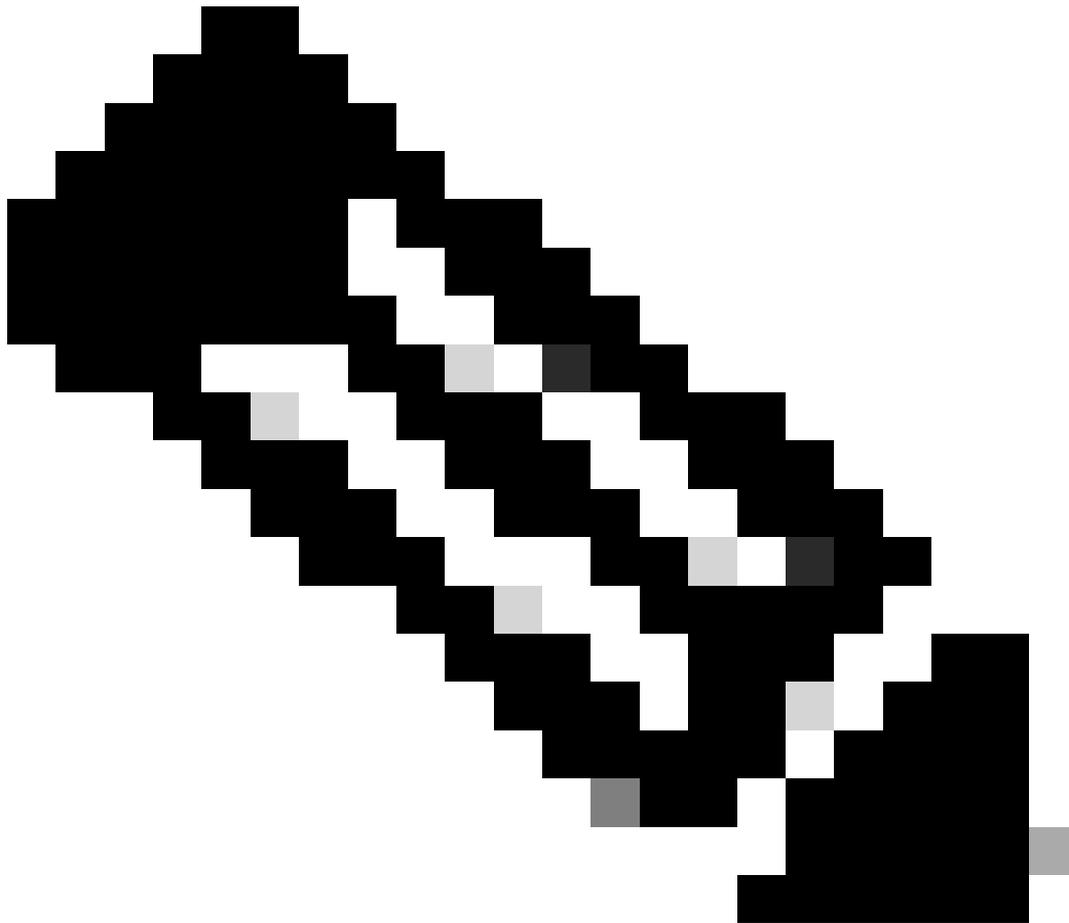
- Anforderungs-Tag:

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

- Hauptkomponenten-Tag:

Dies gilt für die jeweilige Komponente:

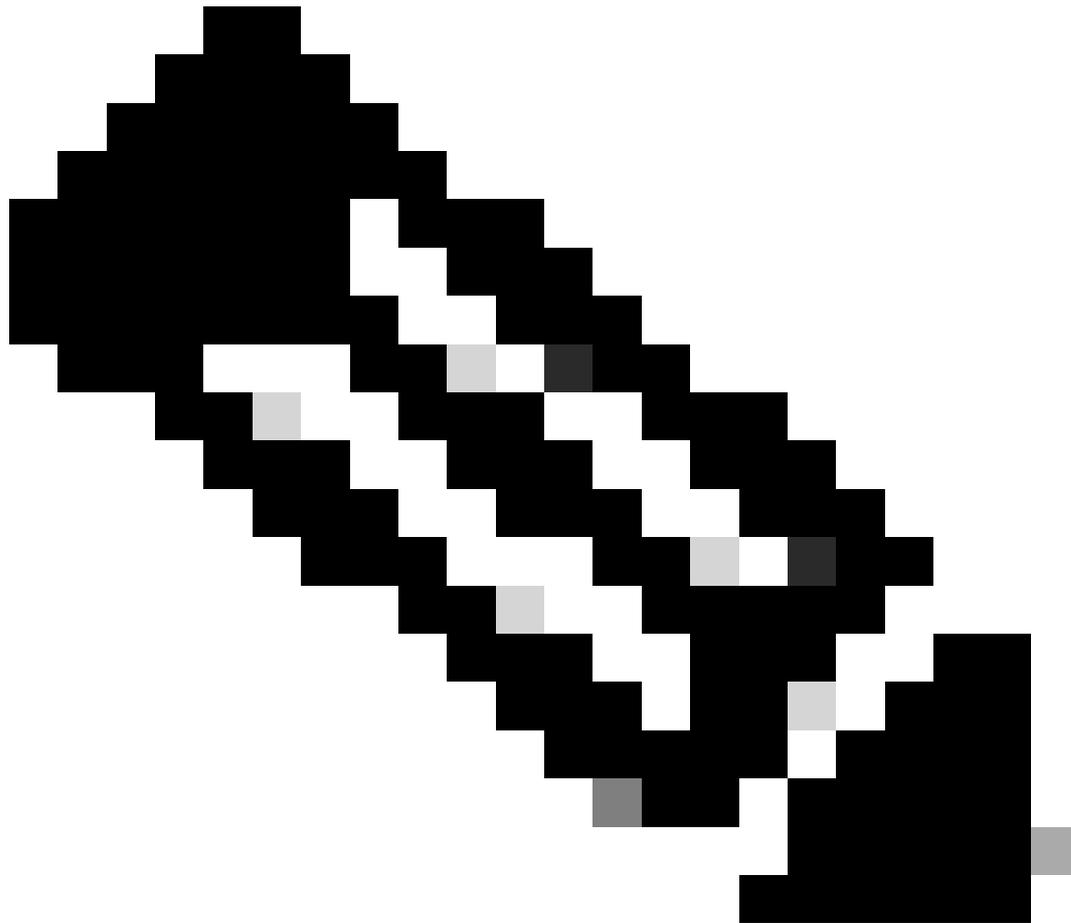
```
<BGP MajorVersion="1" MinorVersion="0">
```



Hinweis: Denken Sie daran, dass Sie die Schemaversion überprüfen können, indem Sie den `GetVersionInfo`-Vorgang auf den Container anwenden.

-
- Kleinere Versionsaktualisierung: Jede Hinzufügung zum XML-Schema, z. B. das Hinzufügen eines neuen Datenelements.
 - Hauptversionsaktualisierung: Semantische Änderungen, Löschungen des Schemas oder der Komponente usw.

Sie können die Version überprüfen. Das nächste Beispiel zeigt, wie man das macht:



Hinweis: Die XML-Vorgänge sind GET, SET usw.

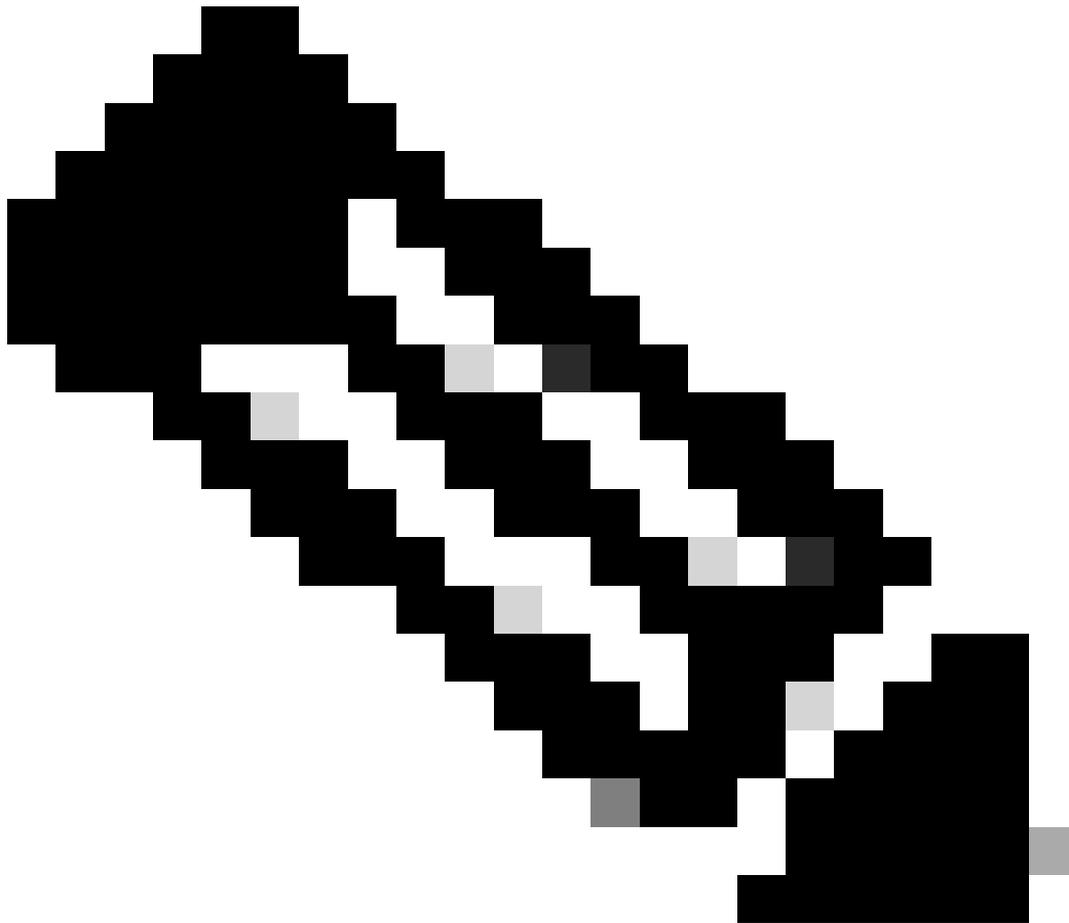
```
<?xml version="1.0" encoding="UTF-8"?>  
<Request Major Version="1" MinorVersion="0">  
<GetVersionInfo>  
<Operation/>  
</GetVersionInfo>  
</Request>
```

Verwenden Sie den Befehl `xml echo formal`, und fügen Sie der Abfrage das Tag `GetVersionInfo` hinzu, um die auf dem Router ausgeführte Version zu finden.

```
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>
```



Hinweis: Request zeigt alle laufenden Versionen der Hauptkomponente sowie die Version in den darin enthaltenen Containern an.

XML-Fehler in Cisco IOS XR

Wenn der Pfad richtig ist, zeigt jede XML-API-Anforderung die angeforderten Informationen an.

Wenn die Abfrage falsch ist

Der Router zeigt drei verschiedene Meldungen an:

- ElementNicht gefunden

Diese Meldung wird jedes Mal angezeigt, wenn eine GET-Operation eine leere Antwort hat.

- ElementNicht gefundenUnten

Der GET-Vorgang enthält diesen Vorgang nicht im XML-Schema.

- Nicht gefunden

Das angeforderte Element kann die Elementebene nicht finden.

Fehlertypen

1. Transport: Fehler in dieser Kategorie schließen alle zwischen der XML-Agent / Client-Kommunikation ein. Dies bedeutet, dass SSH-Interaktionen oder -Probleme beim Transport auftreten können. Um diese Art von Problemen zu überprüfen, wird daher empfohlen, die SSH-Traces zu überprüfen, um nach Problemen mit der Authentifizierung, dem Port usw. zu suchen.

2. XML Parser: Jedes Problem mit dem Format und der Syntax, Probleme in der Antwort gesendet oder in der Abfrage. Diese Probleme senden in der Regel den Fehlergrund, wenn ein Fehler auftritt.

Beispiele:

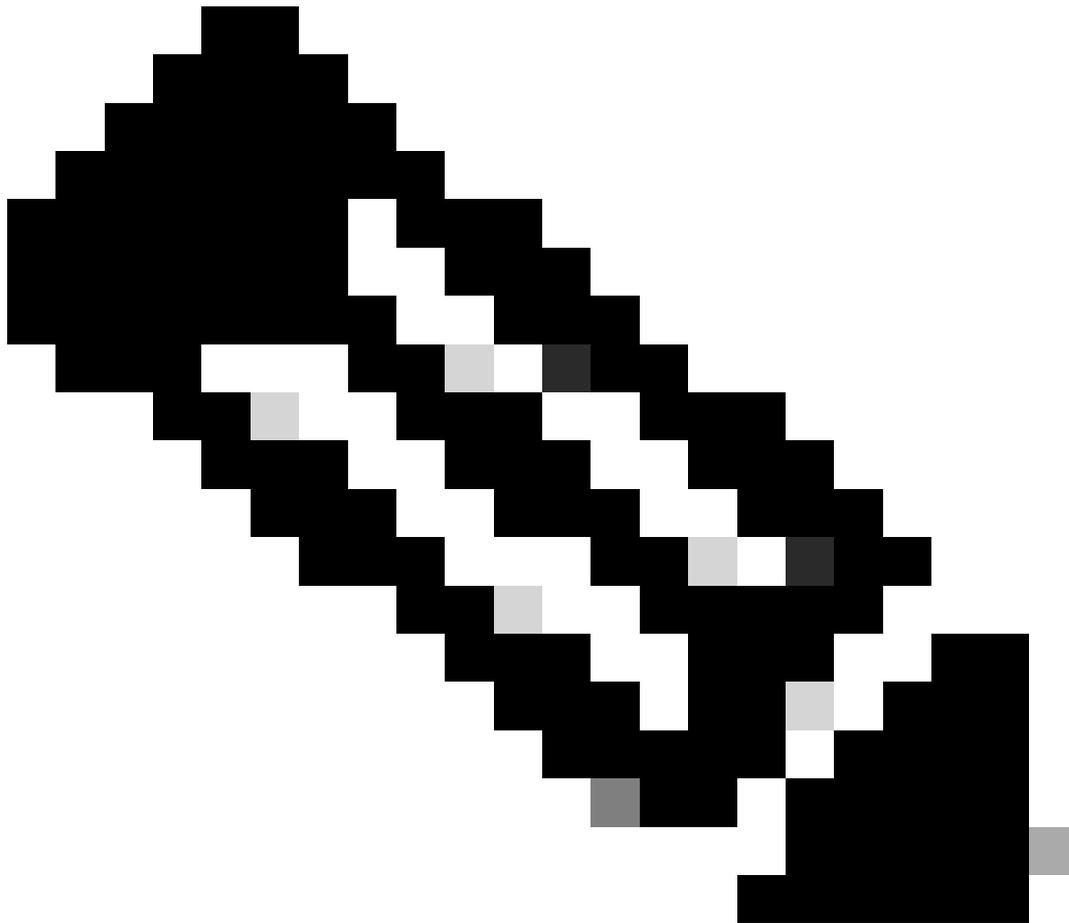
```
ERROR: 0xa367a600 'XML Service Library' detected the 'fatal' condition 'The throttle on the memory usage'
```

3. XML-Schema: Jedes Schema, das sich vom Routerschema unterscheidet. Um diese Probleme zu beheben, überprüfen Sie das CLI-Schema und die CLI-Versionen.

Beispiele:

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

4. Verarbeitung des Vorgangs: Wenn Sie das Gerät konfigurieren, können diese Probleme auftreten. Um dieses Problem zu beheben, müssen Sie die Prozesse wie commit, sysdb usw. beheben.



Hinweis: Fehlerinformationen werden auf der Ebene der Betriebselemente hinzugefügt. Diese wird in Form der Attribute `ErrorCode` (32-Bit int) und `Errormsg` codiert.

Weitere Empfehlungen

Weitere nützliche Techniken:

1. Wildcarding: Dies ist die auch als Spezifische Abfragen bekannt.
2. Batchverarbeitung: Kombination mehrerer Techniken oder Operationen in einer einzigen Anforderung (Best-Effort-Operationen).
3. Benutzerdefinierte Filterung: Wenn das Schema es zulässt, um die Auswahl von Zeilen in Tabellen zu unterstützen.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.