

Cisco IOS XRのXMLのトラブルシューティング

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[スロットルメモリの問題](#)

[問題の詳細説明](#)

[解決方法](#)

[Cisco IOS XRでのXML](#)

[XML APIの使用](#)

[XML APIでサポートされる操作](#)

[提供されるXMLスキーマサービス](#)

[クエリーの構築](#)

[XMLクエリーのテスト](#)

[Cisco IOS XRでのXMLのデバッグ](#)

[APIとクライアントのインタラクション](#)

[Cisco IOS XRでのXMLエラー](#)

[クエリが間違っている場合](#)

[エラータイプ](#)

[その他の推奨事項](#)

はじめに

このドキュメントでは、サービスリクエストおよびCiscoツールに見られるXMLメモリスロットル(TCAM)の問題について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Cisco IOS® XRプラットフォーム
- XML(Extensible Markup Language)インフラストラクチャ
- Common Object Request Broker Architecture(CORBA)
- 要素管理システム(EMS)
- 外部データマネージャ(EDM)

使用するコンポーネント

このドキュメントの情報は、ASR9000に基づくものです。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな (デフォルト) 設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

スロットルメモリの問題

ケースは次の質問でオープンされました。

1. ルータで設定可能な最大メモリは何ですか。
2. サイズの大きいXML応答を分割する方法はありますか。

回答 :

1. 最大メモリは、RSP/ノードが実行しているバージョン (cXRおよびeXR用のバージョン) によって異なります

64ビットバージョン(eXR)スロットルメモリサイズは1つです。

```
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ビットバージョン(cXR)。別の制限があります。

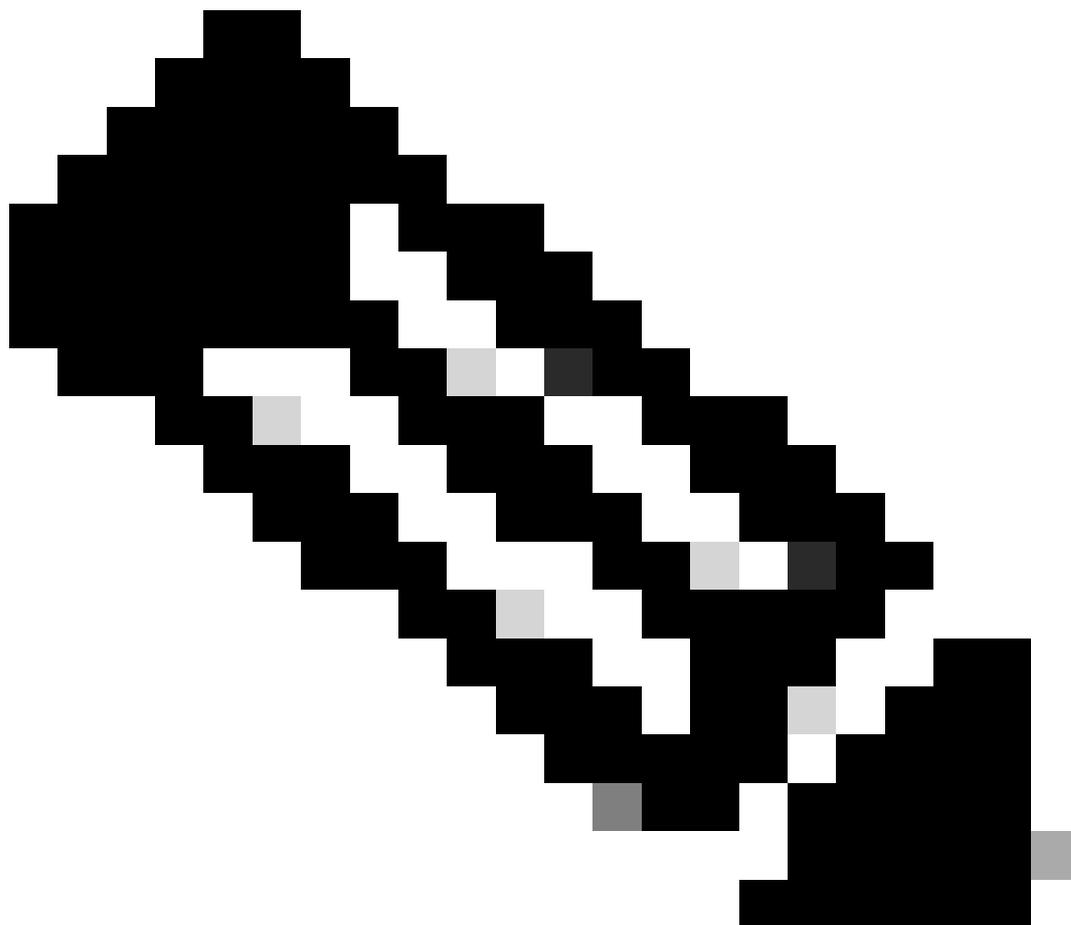
```
RP/0/RSP0/CPU0:XR#show version
Cisco IOS XR Software, Version 6.4.2[Default]
```

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：どちらのバージョンでも、デフォルトは300 MBです

2. XML応答を中断する方法はいくつかあります。

- イテレータ：

イテレータを設定するときは、特定のウィンドウサイズでXML合計応答をセグメント化しま

す。ウィンドウにはイテレータサイズがあります。たとえば、応答が1 Gbで、イテレータサイズが500 Mbの場合、XMLを2つの応答にセグメント化します。

このメソッドは、Iteratorのサイズに基づいてGetNext操作（応答が必要とするもの）を追加することにより、応答を変更します。

1つのセッションに対して最大10のイテレータを設定できます。

- スロットル（プロセスレート）:

この機能は、XMLプロセスによって消費されるメモリを制限します。プロセスがメモリをオーバーパスした場合、プロセスは「メモリ使用量のスロットルに達しました」というエラーメッセージを返します。

- ストリーミング:

イテレータと同様に、ストリーミングはXML応答を特定のウィンドウにセグメント化します。違いは応答で、GetNext操作とイテレータIDが削除されます。XMLはストリームされたパッケージを送信し、クライアントはストリームが終了したときに応答を作成します。

問題の詳細説明

自動化が必要な使用例では、[pyIOSXR](#)ツールを使用します。この自動化ツールはXMLエージェントで、いくつかのshowコマンドを発行し、全体としてデバイスに接続するのに役立ちます。

このエージェントで大きな要求を送信するたびに、エラーが表示されます。

注：シスコの内部ツールおよび情報にアクセスできるのは、登録ユーザのみです。

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

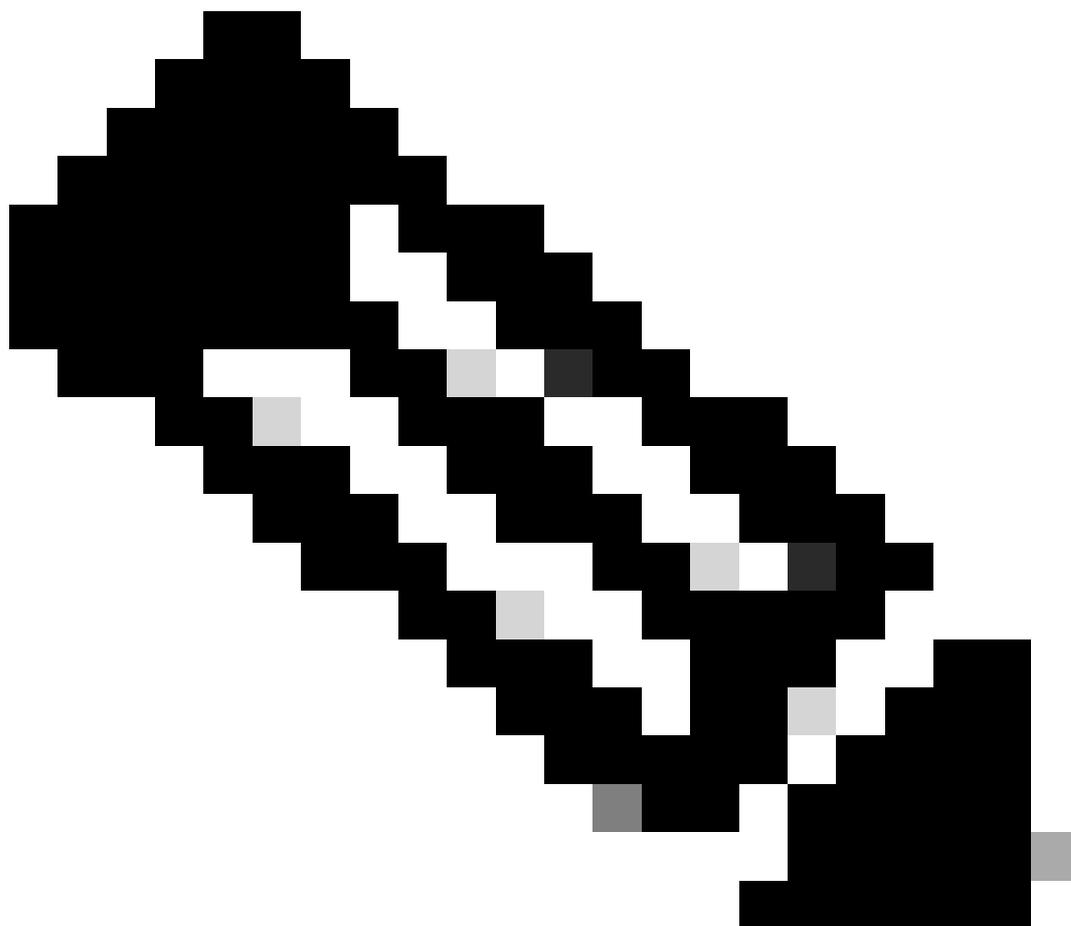
「リクエストを最適化して小さいデータを照会する」というエラーメッセージが表示されたら、イテレータを使用してエージェントXML APIを設定できます。理論上、これは応答のセグメント化を可能にします。

イテレータを設定すると、同じエラーメッセージ「RESPONSE ERROR...Please optimize the request to query small data」が表示されます。

エラーが表示された場合は、次の手順として、この場合はイテレータの機能がクエリーに対して機能しなかった理由を理解します。

[pyIOSXR](#)では、XMLエージェントをデバイスで正しく有効にすることをお勧めします。これは、APIでイテレータの使用が許可されていないことを意味します。

次の手順は、2番目のオプションであるストリーミングをテストすることです。



注: pyIOSXRでは、基本的なXMLクエリ以外の異なるヘッダーを使用して次の要素を処理することはできません。

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

ストリーミングとイテレータは、クエリにヘッダーを追加します。ストリーミングは、システムが完全な応答を構築するのに役立つストリームIDを追加します。イテレータはGET_NextとイテレータIDを追加します。

ストリーミングのテストでも、イテレータと同じエラーメッセージが表示されました。

解決方法

「デバイスに対するクエリの実行に関する推奨事項の追加」セクションには、これらの問題を分類するためのより多くのツールが示されています。ワイルドカードもその1つです。ワイルドカードは、メモリスロットル制限の解決策です。

ワイルドカードは、不要な情報を要求しないように特定のクエリを作成します。たとえば、BGP情報を表示するには、一般的なshow routeコマンドの代わりにshow route bgpコマンドを使用します。この例は、XMLクエリとロジックに適用されます。システムにバルク情報を要求すると、メモリと処理の問題が発生する可能性があります。

ケースが開かれたとき、このクエリが使用されました：

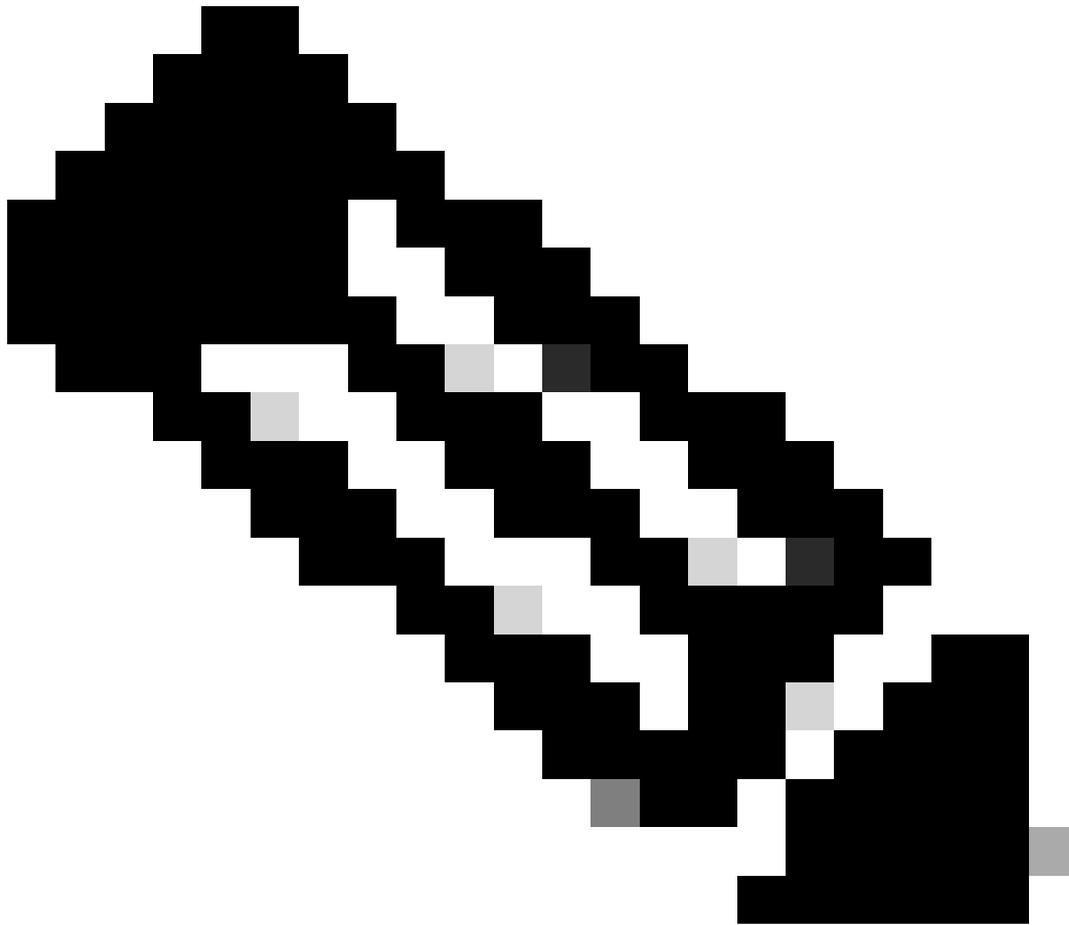
```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get>
    <Operational>
      <BGP MajorVersion="36" MinorVersion="1">
        <InstanceTable>
          <Instance>
            <Naming>
              <InstanceName>
                DEFAULT
              </InstanceName>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

このクエリは、フルBGPテーブルを対象としています。この非固有の要求に対する応答は2.2 Gbに近かったため、メモリのスロットル制限に達しました。

これを修正するには、特定のクエリが必要です。これにより、システムはクエリを処理し、情報を返すことができます。

Cisco IOS XRでのXML



注：詳細な情報、仕様、またはその他のコマンドが必要な場合は、次のリンクをクリックしてください。XMLプログラミングガイド：[XR XMLプログラミングガイド](#)

XMLは、データの表示と構造化の方法を定義します。これは、コンピュータがビットとして認識する内容を解析し、構造化された標準化された情報を表示する方法です。

XMLの構造は次のとおりです。

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

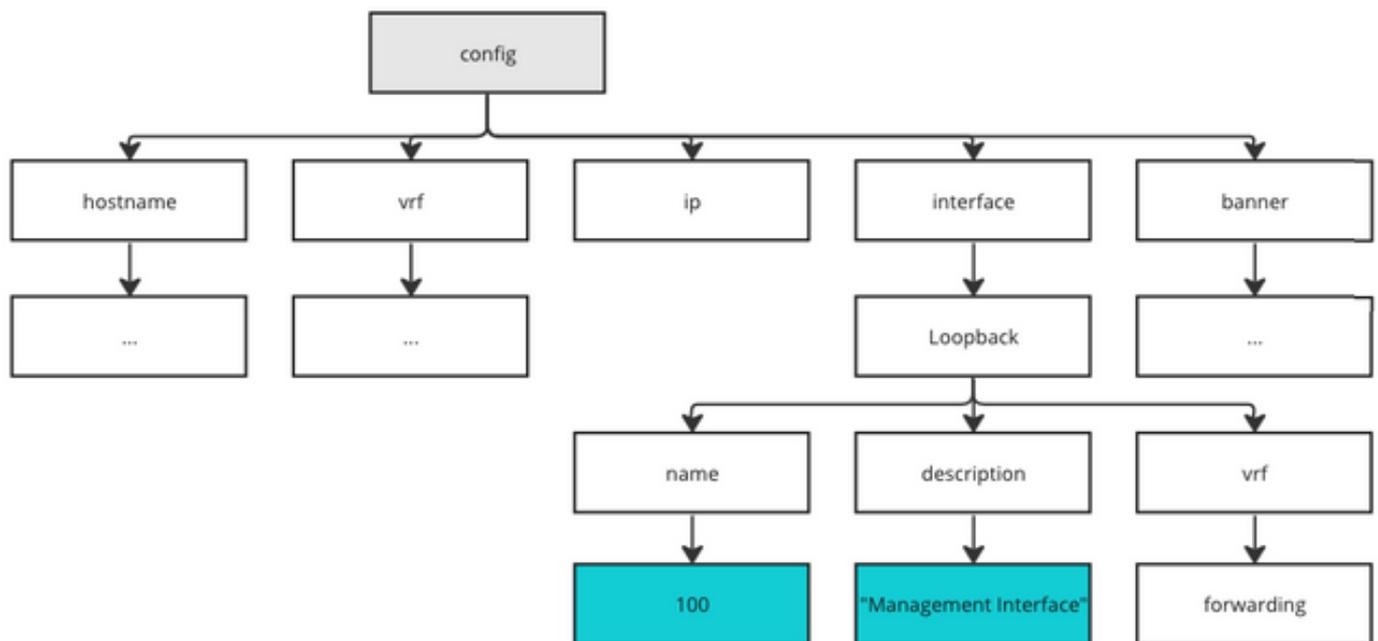
すべてのタグには、開始タグ(<init>)と終了タグ(</init>)の2つの部分があります。この構造が使用されていない場合、XMLはタグが終了する場所を理解できません。

XMLは次の2つのエンティティで構成されています。

- コンテナ
- リーフ

この構造化されたデータに類似しているのは、XMLデータをツリーとして比較することです。コンテナはブランチであり、ブランチごとにスタブされたリーフがあります。リーフには情報以外の情報は含まれません。

たとえば、次のツリーのルートはグレー、コンテナは白、リーフは青です。



XML APIの使用

XML APIをテスト/利用するには、まずクエリが必要です。

1. クエリーにヘッダーがある場合：

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



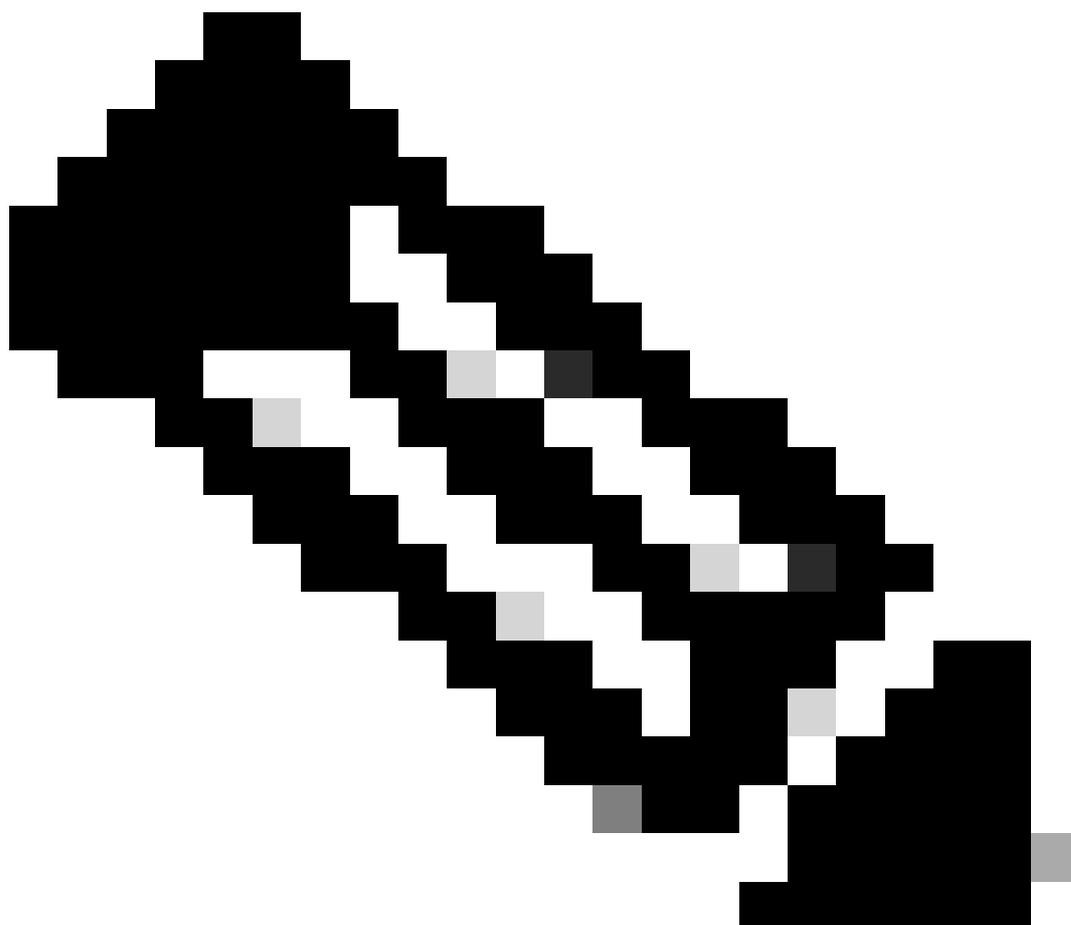
注：これは、終了タグを必要としない唯一のタグです。

2. 要求のタグを追加する必要があります。要求では、バージョンを指定する必要があります。

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

3. ヘッダー、要求およびバージョンを使用して、本体はAPIが持つすべてのXML操作を続行します。

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



注意：要求の問合せごとに異なる操作が含まれていることに注意してください。操作ごとに要求を実行する必要はありません。

次の例では、必要なすべての情報を含む要求を表示します。

```
<?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でサポートされる操作

Cisco IOS XRは、ユーザがXMLスキーマの持つ情報を操作できる5つの操作をサポートしています。

1. ネイティブ・データ・オペレーション：

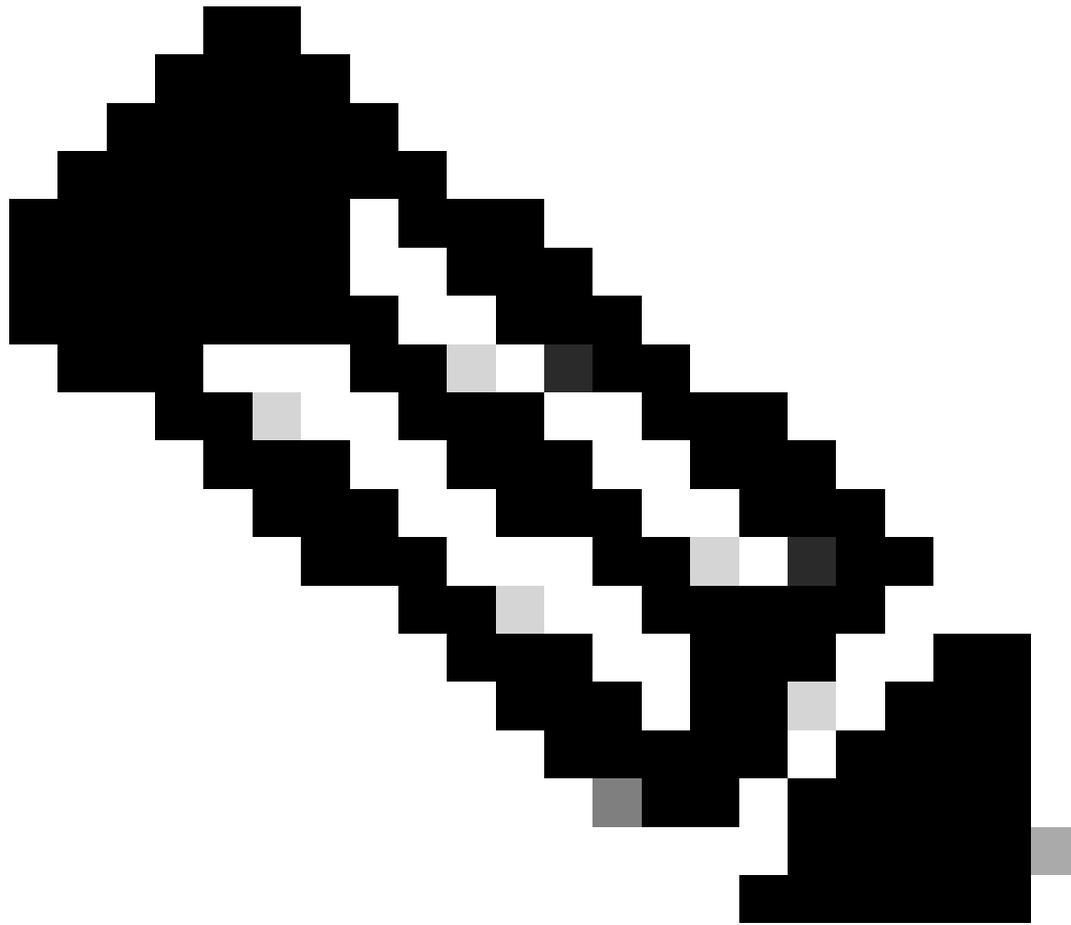
- <Get></Get>：アクションデータ項目（リーフが持つデータ）を収集します。
- <Set></Set>：データ項目を構成、変更、または追加します。
- <Delete></Delete>：1つ以上のデータ項目を削除します。
- <GetVersionInfo></GetVersionInfo>：要求されたコンポーネントのメジャーバージョン番号とマイナーバージョン番号を取得する操作です。
- <GetDataSpaceInfo></GetDataSpaceInfo>：この操作タグは、コンテナにマッピングされたリーフ名を表示します。

2. CLI操作：

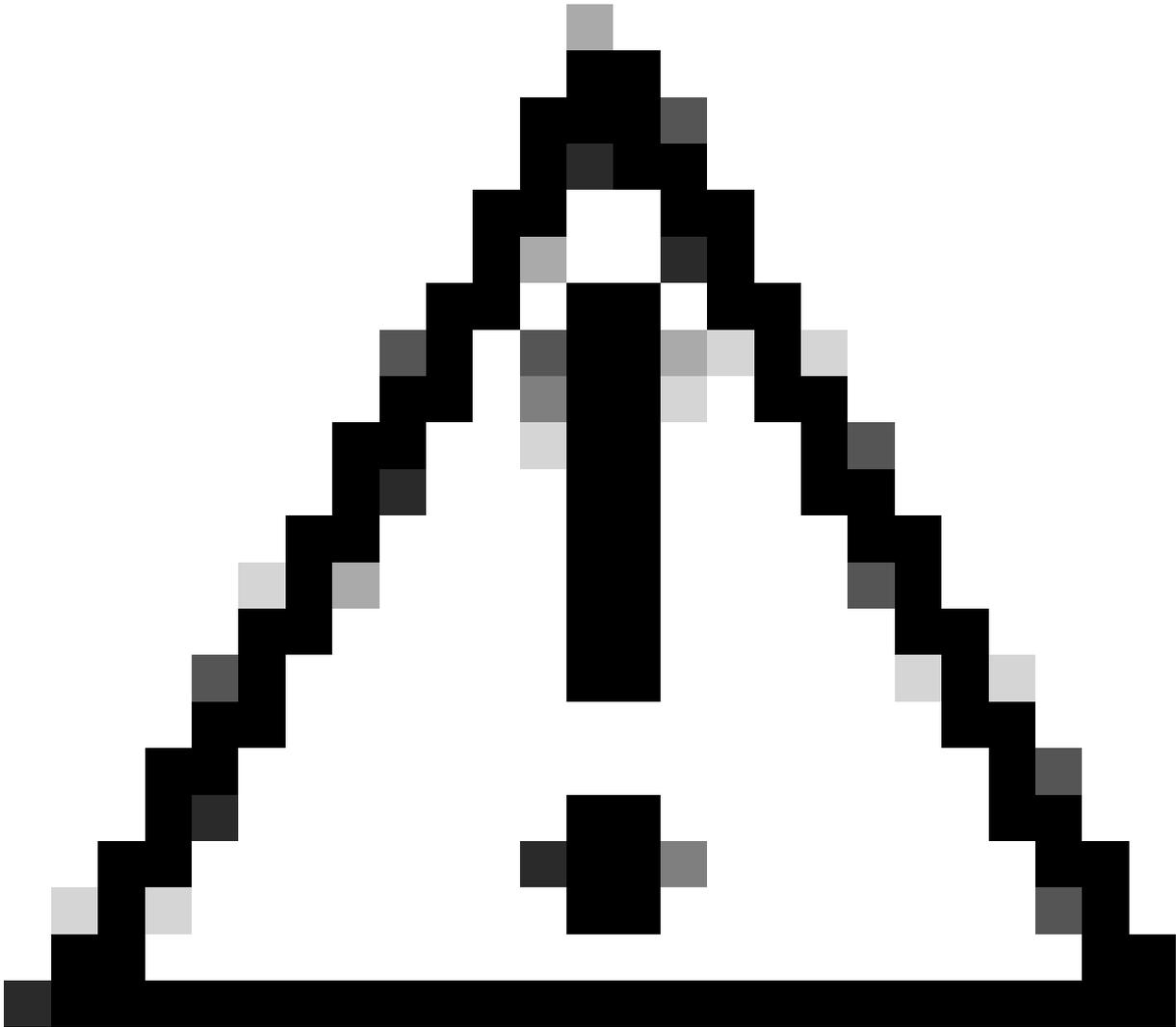
- <CLI></CLI/>：CLIコマンドとして設定要求を発行できるタグ。
- <GetNext></GetNext/>：データがブロックサイズより大きい場合、このタグは次にセグメント化された応答に追加されます。

提供されるXMLスキーマサービス

XML APIは、次の名前空間を実装します。



注：ルートオブジェクトを表示する必要がある場合は、[Cisco IOS-XR XML Network Management](#)を参照してください。



注意：名前空間ごとに異なるデータと操作を管理する点に注意してください。

-
- 設定：Get、Set、およびDeleteアクションを許可する操作。
 - 動作可能：CLI showコマンドと同様。
 - アクション：設定アクションをサポートします。この操作により、clearコマンドやその他の類似コマンドなどのアクションデータにアクセスできます。
 - AdminOperational：管理操作データへのアクセス。
 - AdminAction：管理アクションデータへのアクセス。
 - AdminConfiguration：管理構成データへのアクセス。

クエリーの構築

移動には2つの異なるパスがあります。クエリの名前空間を選択したら、スキーマを使用して次の操作を行う必要があります。

1. XMLスキーマのドキュメント

これらのオプションは、コンテナとリーフを表示する完全なXMLツリーを提供します。ドキュメントについては、[CRS XML Schemas](#)をクリックしてください。

ドキュメントページが表示されます。

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

このページには、オブジェクト、コンテナおよび子が表示されます。各オブジェクトには子オブジェクトが含まれます。子オブジェクトに他のコンテナが含まれていない場合、子オブジェクトはリーフと見なされます。

子をクリックすると、クエリーを作成するための情報が表示されます。IPv4の例：

<IPV4>

Description: IPV4 related services

Task IDs required: ipv4

Parent objects:

[<Services>](#)

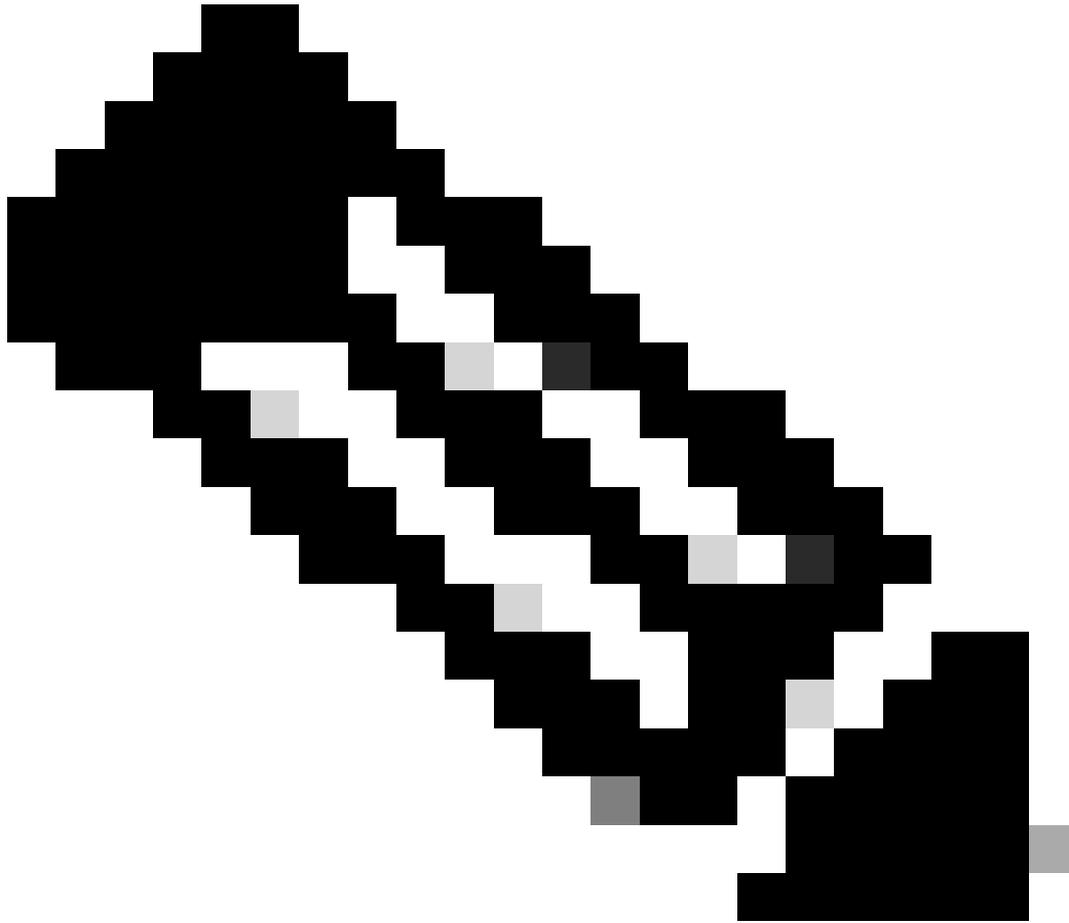
Child objects:

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

Available Paths:

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

前の図では、すべてのIPv4クエリに対して設定操作を実行できます。コンテナIP、コンテナCinetd、コンテナServices、そしてリーフIPv4をターゲットにします。



注：クエリを実行するためにリーフにする必要のある操作。

クエリは次のようになります。

```
<?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

XML APIはCisco IOS XRに組み込まれています。XMLツリーを表示するには、show xml schemaコマンドを使用します。コマンドを発行すると、Linuxに似たCLIでXMLスキーマ全体が表示されます。

このCLIでは、次のアクションがあります。

? : まるで—help

LS: ブランチユーザのコンテナまたはリーフをリストします。

<operation>:?を発行すると、操作をチェックでき、コマンドを発行できます。

次のような操作を行います。

-設定

- adminconfig (オプション)

-cd

-list

- オペア

-action

CD : ディレクトリ/コンテナの変更

次の例を示します。

```
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：デバイスにログインするにはクレデンシャルが必要です。これらのクレデンシャルはデバイスに対してローカルであり、root-Is/adminプロファイルが必要です。

次の例では、XML APIを使用してクエリを作成する方法を示します。たとえば、クエリはXMLエージェントが有効かどうかを確認する必要があります。

```
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
```

次のプロンプトに注目してください。

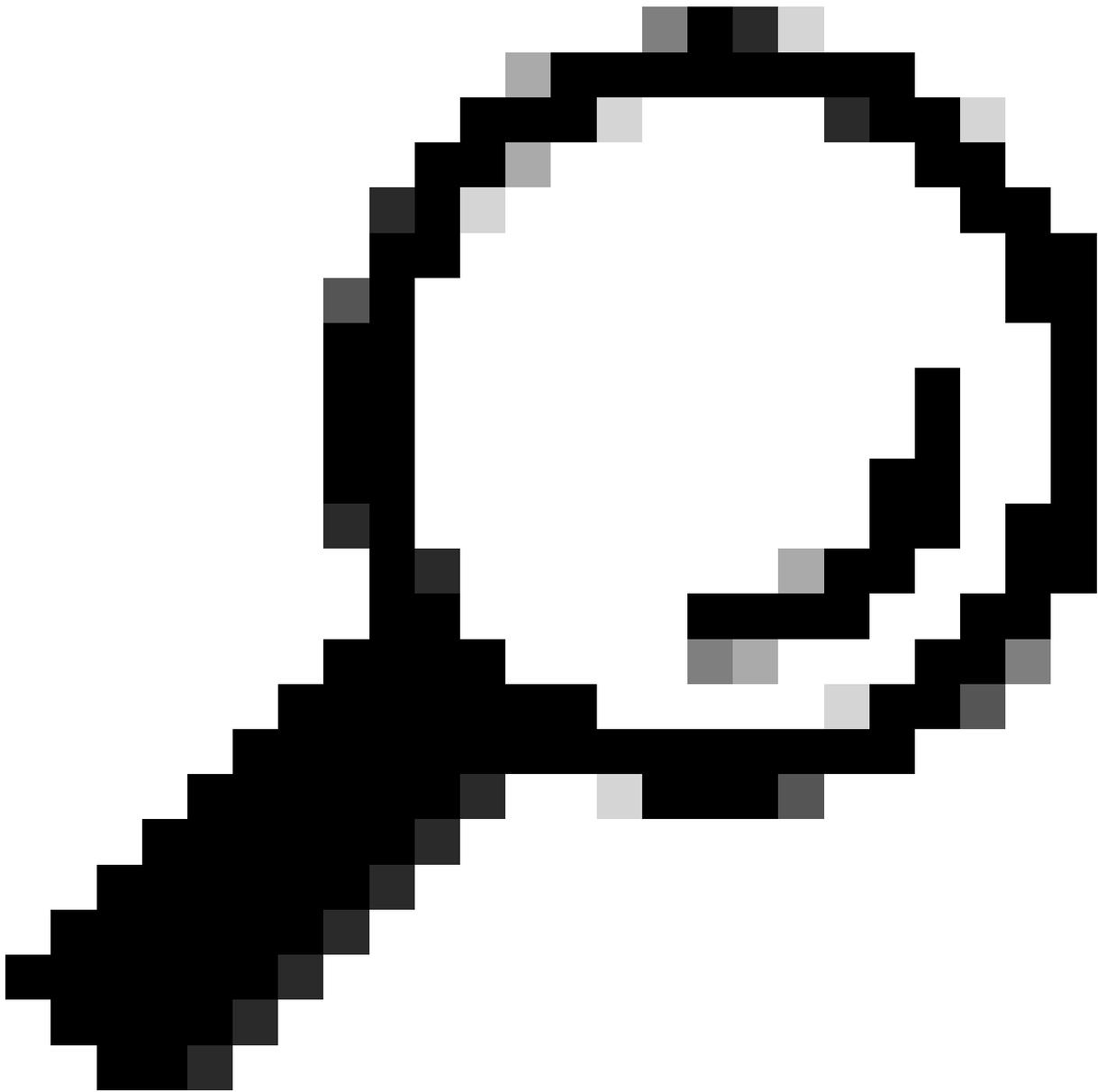
```
xml-schema[config]:XML.Agent.TTY>
```

前のプロンプトにXMLツリーが表示されます。この情報を使用して、クエリーは次のようになります。

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

XMLクエリーのテスト

クエリーが構築されたら、次の手順はそれをテストすることです。これをテストするには、CLIで `xml echo format` コマンドを発行します。このアクションは同じデバイスにアーカイブできます。



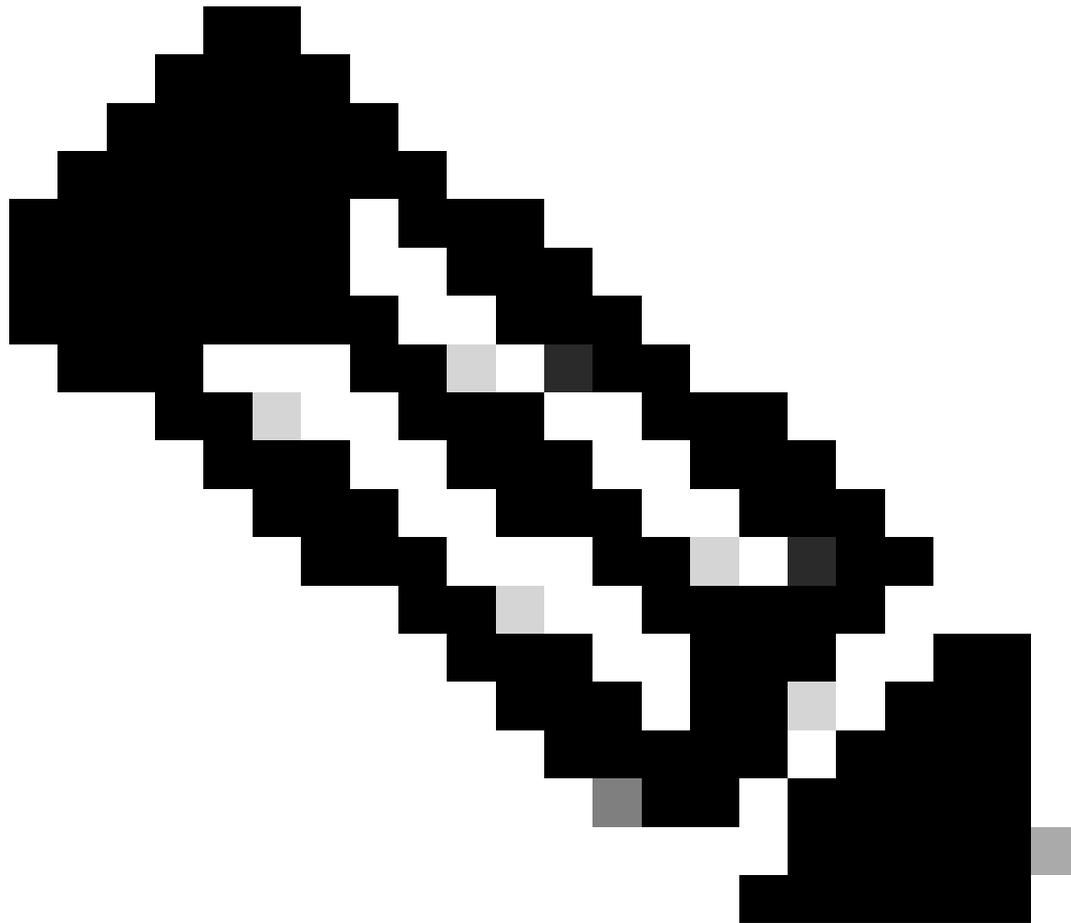
ヒント：前のコマンドを入力した後で、ユーザがクエリをコピー/ペーストしてEnterキーを押すことができることを伝えてください。デバイスはクエリに対する応答を表示します。

```
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
```

ルータは応答とResultSummaryを表示します。



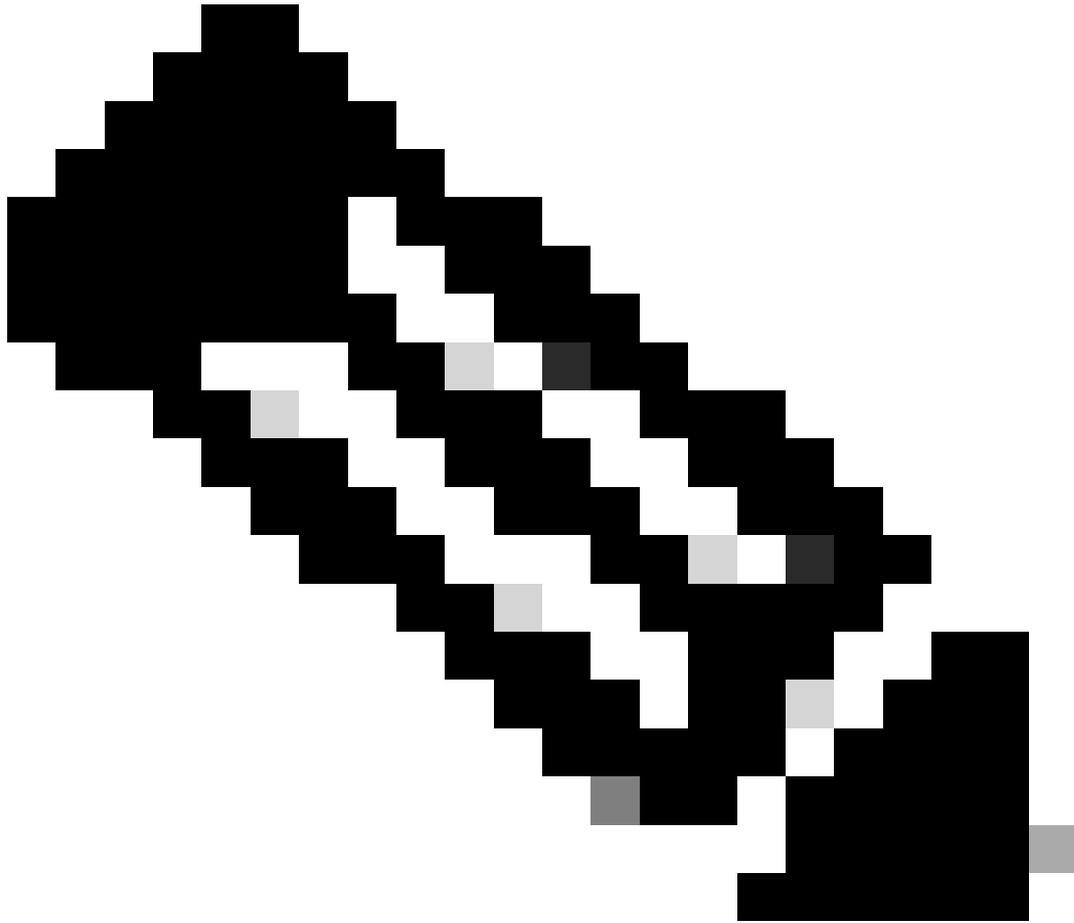
注：このXML応答では、スロットルメモリを表示したり、設定したりする必要はありません。応答は、SSH/TTYを使用していないため、サイズに関係なく要求されます。1つはXMLクライアントで、もう1つは埋め込みXML echoコマンドです。動作は変更される可能性があります。

Cisco IOS XRでのXMLのデープダイブ

Cisco IOS XRは、動作するために異なるパッケージを必要とするモジュラシステムです。XMLは管理プレーンサブシステムにあると見なされます。

XRが開始した時点では、デバイスでXMLを使用する方法は2通りありました。

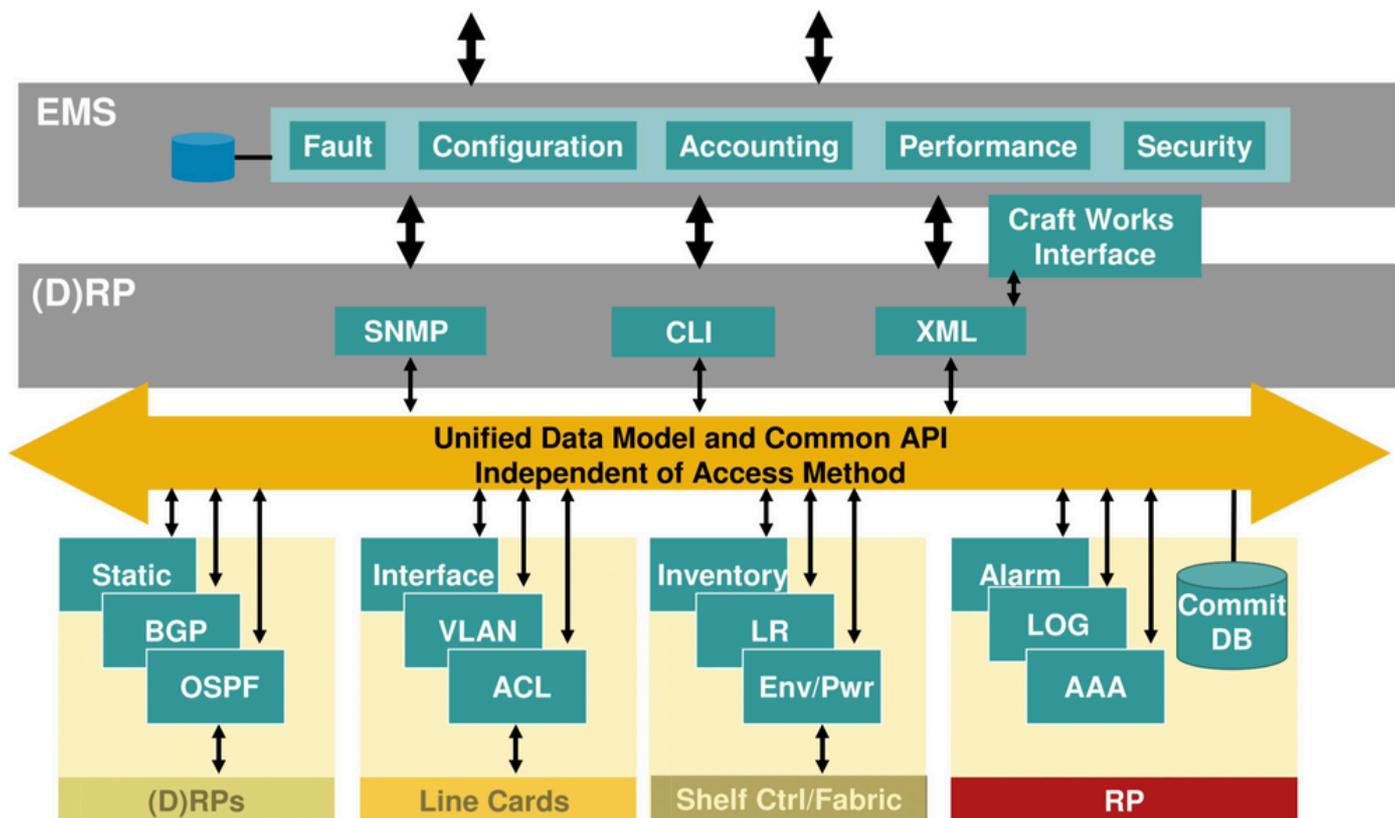
1. CORBAエージェント (SSLを使用)
2. SSHエージェント。



注:CORBAはバージョン3.7以降で廃止されています。

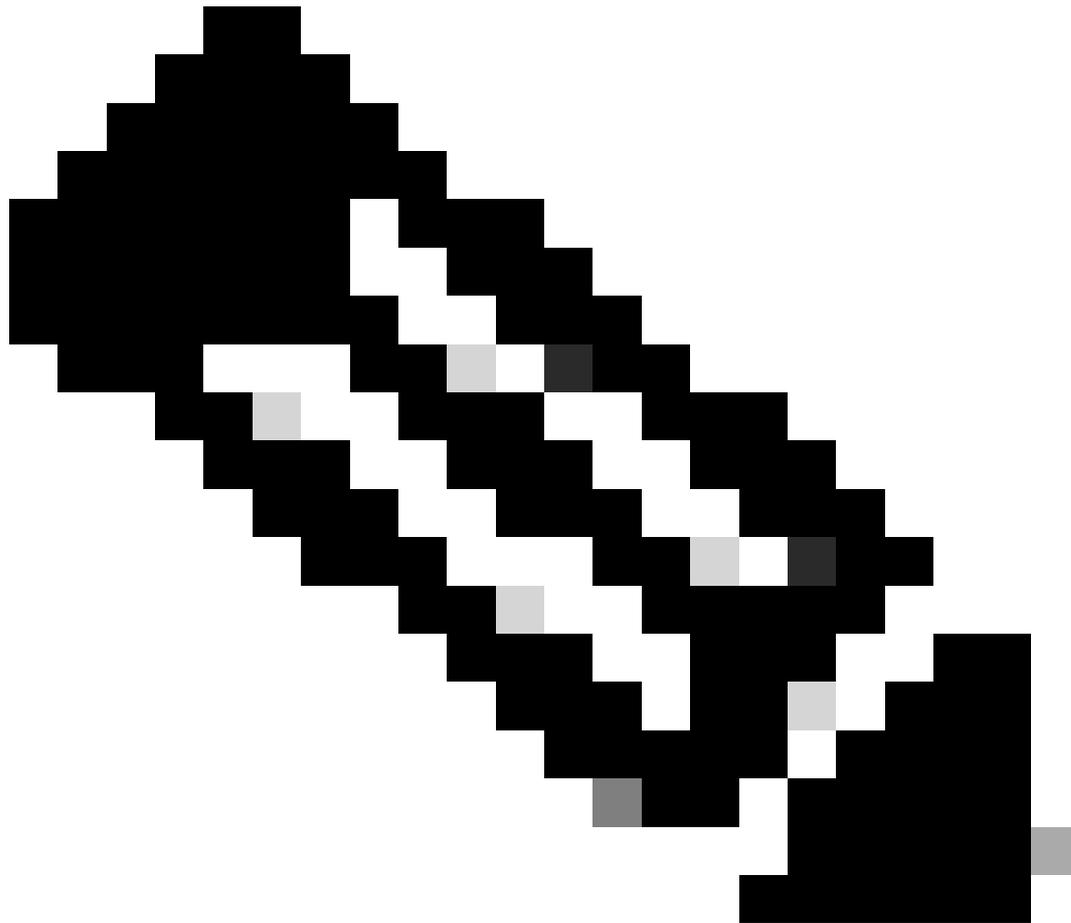
CORBAは、セキュリティパッケージにあるSSLを使用するレガシープロトコルです。一方、SSH方式では、Manageability Packageを使用する必要があります。

インフラストラクチャは次のように分散されます。



前の図から、XMLの主なプロセスはルータおよびスイッチプロセッサ(RSP)で見られます。これらのプロセスには、デバイス内の他のプロセスからのすべての情報を監視する共通のAPIがあります。これらのプロセスは次のとおりです。

- ラインカードから :
 - 統計コレクタ
 - 統計サーバ
- ルータプロセッサから :
 - [統計マネージャ]:
 - SysDBです。
 - TCP(XML APIはTCP 38751を使用)
 - Netio

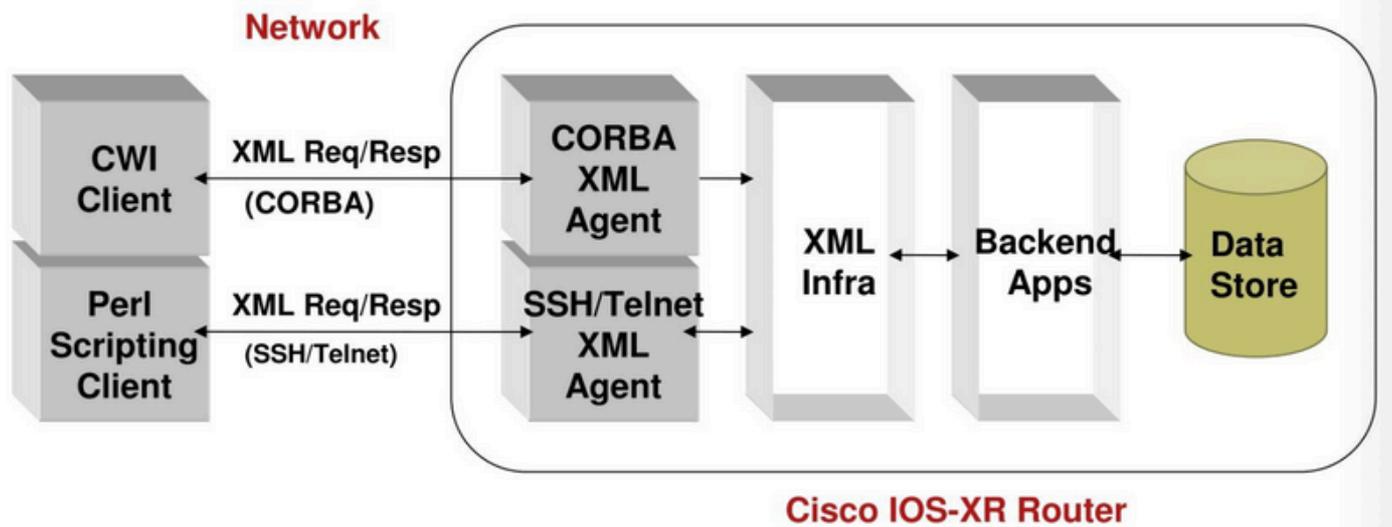


注：これに関する詳細は、「Cisco IOS XRのXMLエラー」セクションを参照してください。

要求に応じて、1つ以上のプロセスがトリガーされます。自動化ツールを開発する際に、応答がブロックサイズよりも大きい場合、応答は成功しません。プロセスが応答に時間がかかる場合、EDMログを生成したり、シャットダウンしたり、サービスに影響を与えたりすることがあります。

APIとクライアントのインタラクション

次の図は、インタラクションを示しています。



デバイスでXMLエージェントを設定または有効化するには、次のコマンドを使用します。

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

システムは、次の領域でクライアントと合意する必要があります。

1. バージョン
2. スキーマ

バージョンは、次の2つの領域で定義できます。

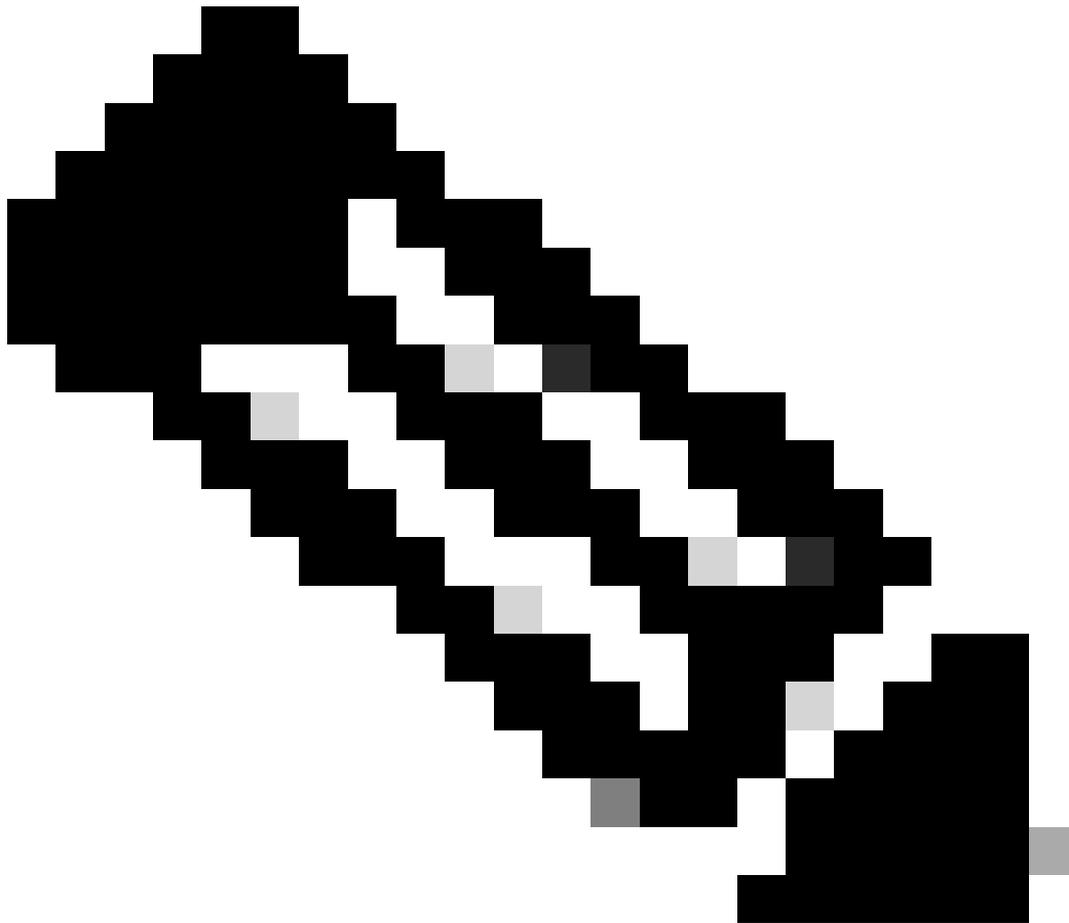
- 要求タグ :

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

- 主要コンポーネントタグ :

これは特定のコンポーネントに適用されます。

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



注：コンテナに適用されたGetVersionInfo操作を使用して、スキーマのバージョンを確認できます。

-
- マイナーバージョンの更新：新しいデータ項目の追加など、XMLスキーマへの追加。
 - メジャーバージョンの更新：セマンティックの変更、スキーマまたはコンポーネントの削除など。

バージョンを確認できます。次の例は、その方法を示しています。

注:XML操作には、GET、SETなどがあります。

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

xml echo formalコマンドを使用し、クエリにタグGetVersionInfoを追加して、ルータで実行されているバージョンを確認します。次に例を示します。

```
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>
```

注意：要求では、主要コンポーネントで実行されているすべてのバージョンが表示され、その中のコンテナにもバージョンが表示されます。

Cisco IOS XRでのXMLエラー

パスが正しければ、各XML API要求は要求された情報を表示します。

クエリが間違っている場合

ルータは、次の3つの異なるメッセージを表示します。

- アイテムが見つかりません

このメッセージは、GET操作に空の応答があるたびに表示されます。

- ItemNotFound以下

GET操作にXMLスキーマにこの操作が含まれていません。

- 見つかりません

要求された要素は、要素レベルを見つけることができません。

エラータイプ

1. Transport : このカテゴリのエラーには、XMLエージェント/クライアント間の通信に関するエラーが含まれます。これは、トランスポートでSSHのインタラクションや問題が発生する可能性があることを意味します。したがって、このような種類の問題を確認するには、SSHトレースを確認して、認証やポートなどの問題がないかを確認することを推奨します。

2. XMLパーサー : 形式や構文に問題がある、送信された応答や問い合わせの問題。通常、これらの問題は、エラーが発生したときに失敗の理由を送信します。

例 :

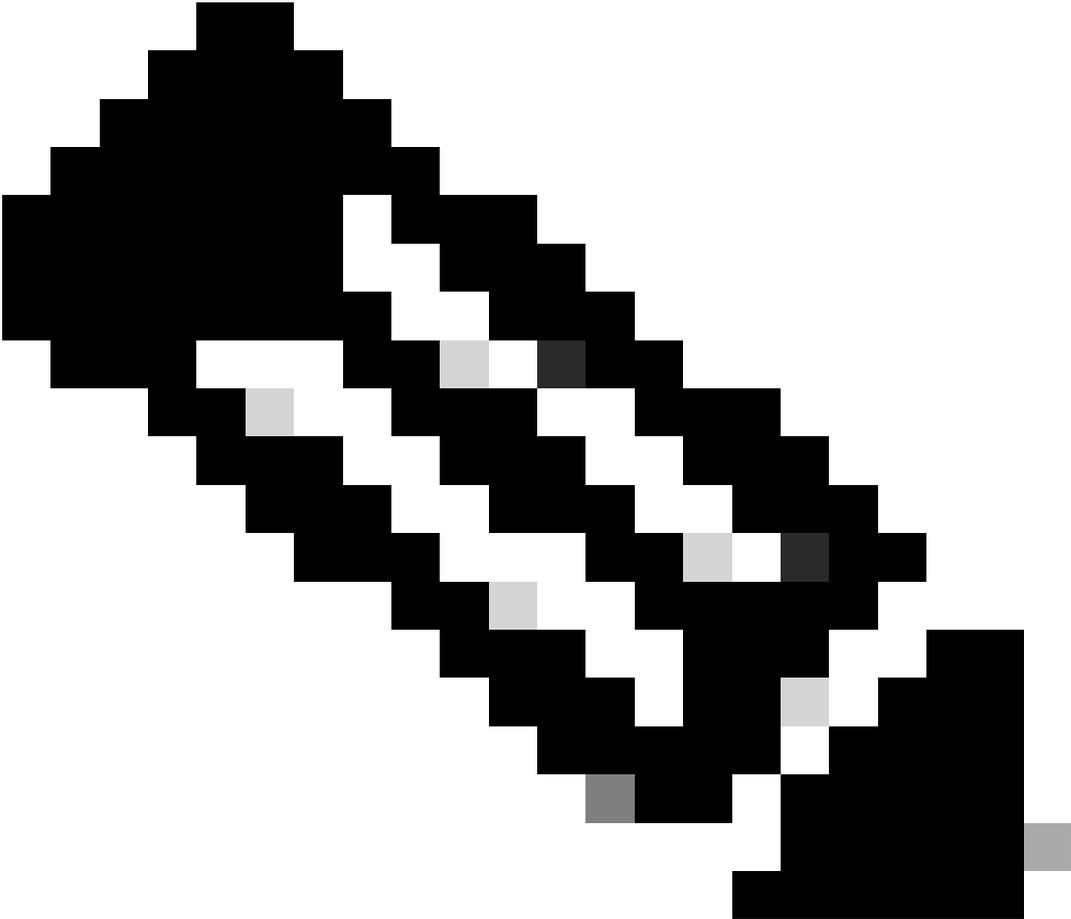
```
ERROR: 0xa367a600 'XML Service Library' detected the 'fatal' condition 'The throttle on the memory usage'
```

3. XMLスキーマ : ルータスキーマとは異なるスキーマ。これらの問題に対処するには、CLIスキーマとバージョンを確認します。

例 :

```
ErrorCode="0x43688400" ErrorMessage="'XMLMDA' detected the 'warning' condition 'An error occurred while processing the request'"
```

4. 操作処理 : デバイスを設定すると、次の問題が発生する可能性があります。この問題を解決するには、commit、sysdbなどのプロセスをトラブルシューティングする必要があります。



注意：エラー情報は工程要素レベルに追加されます。これはErrorCode (32ビットint) と Errormsg属性の形式でコード化されます。

その他の推奨事項

その他の便利なテクニック：

1. ワイルドカードインギ：これは特定クエリとも呼ばれます。
2. バッチ処理：複数のテクニックまたはオペレーションを1つのリクエストに結合します (ベストエフォート型オペレーション) 。
3. カスタムフィルタリング：スキーマが許可する場合、テーブル内の行の選択を支援します。

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。