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
- クエリーの作成
- XMLクエリーのテスト
- Cisco IOS XRでのXMLのディープダイブ
- 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
- クエリが正しくない場合
- 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

次の質問を含むケースがオープンしました。

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応答を破棄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のバージョン）に応じて変化します

64ビット版(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)では、 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]
```

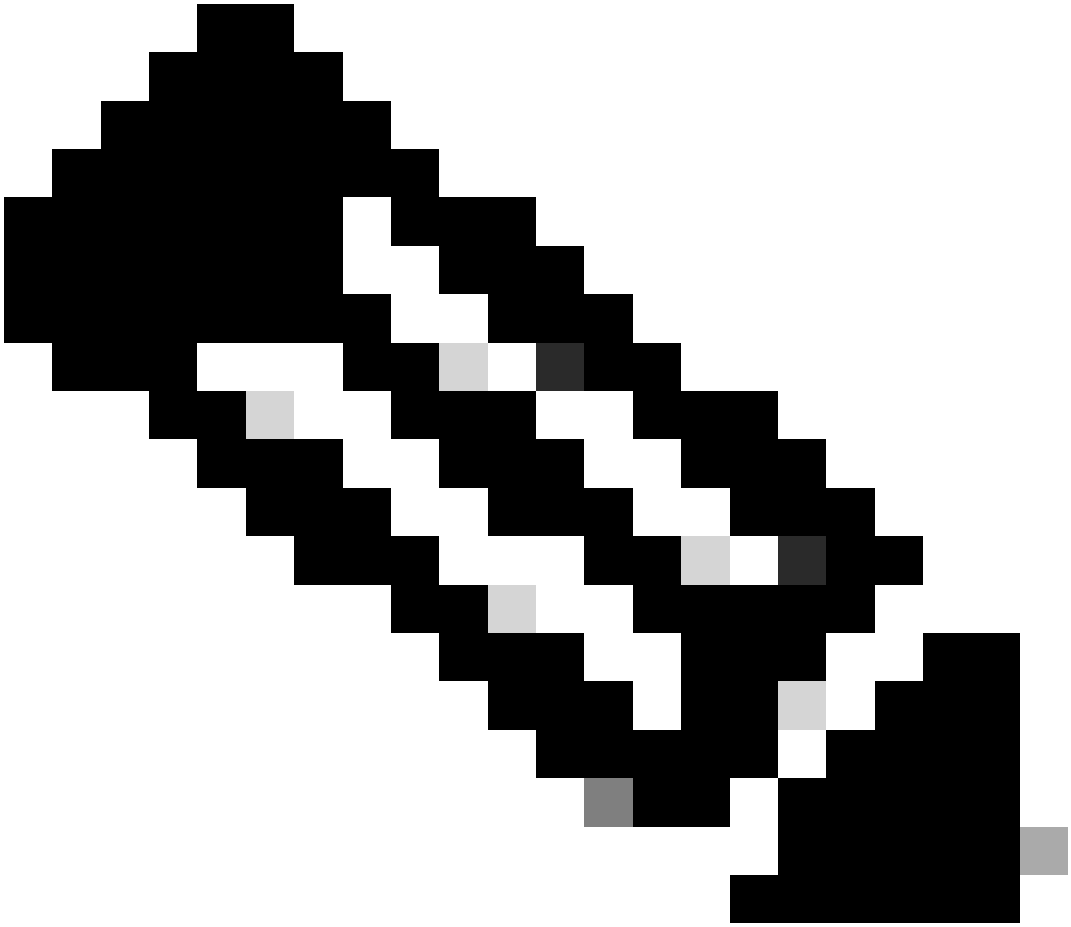
```
Copyright (c) 2020 by Cisco Systems, Inc.

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




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合計応答のセグメント化を行います。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、（イテレータサイズに基づいて）GetNext操作（応答が必要とするもの）を追加することで、応答を変更します。

  1つのセッションに対して最大10のイテレータを指定できます。

- スロットリング(process-rate):

  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超過した場合、プロセスはエラーメッセージ「The throttle on the memory usage has been reached」を返します。

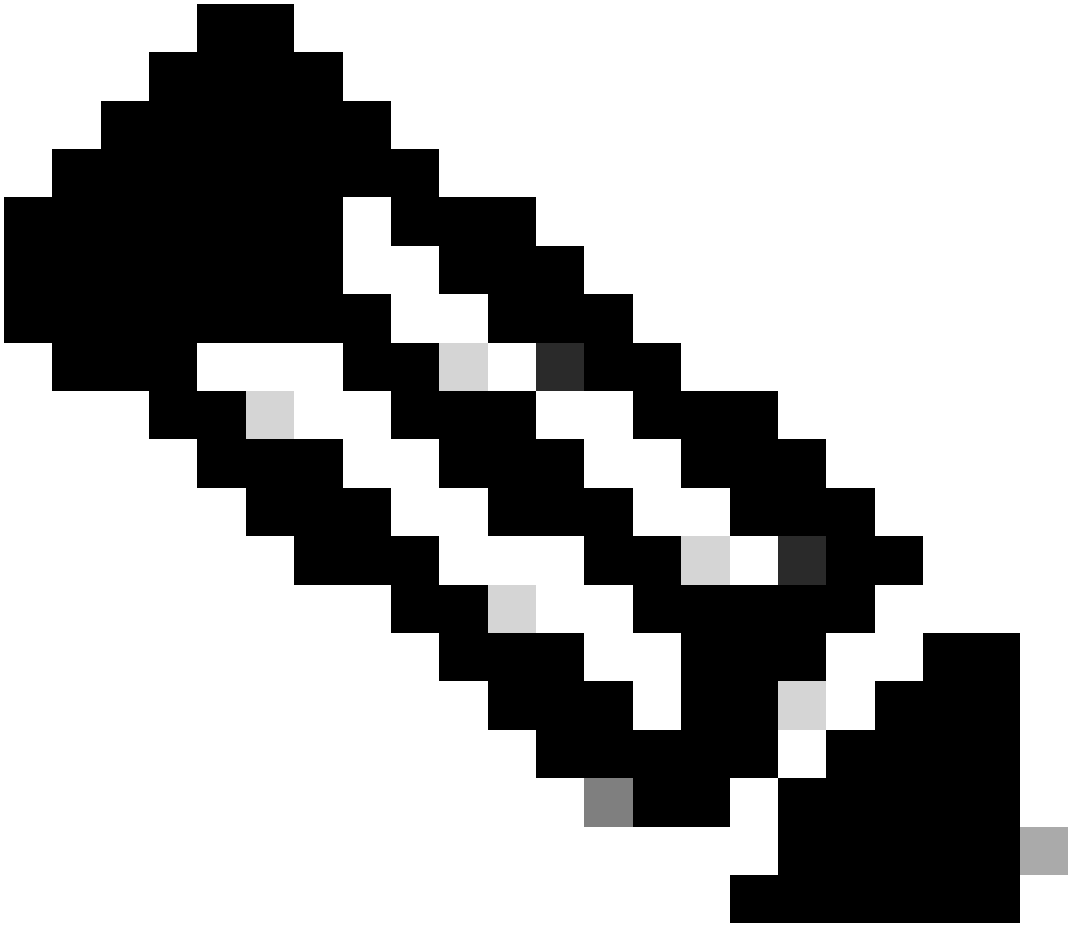
- ストリーミング：

  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。違いは応答であり、GetNext操作とイテレータIDを削除します。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、登録済みのシスコユーザーのみです。

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

「optimize the request to query smaller data」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。

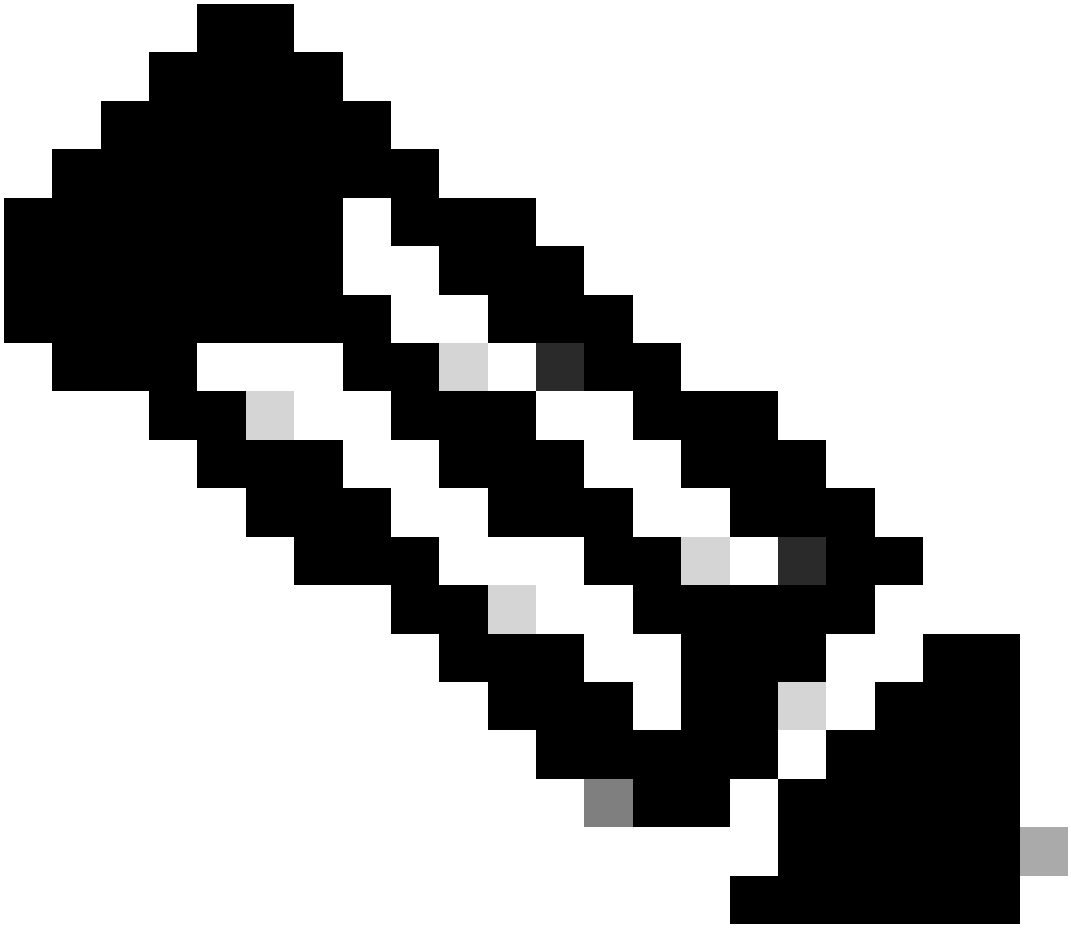
イテレータを設定すると、同じエラーメッセージ「RESPONSE ERROR...Please optimize the request to query smaller data」が表示されます。

エラーが表示された場合は、次の手順として、その機能（この場合はイテレータ）がクエリ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化することを推奨しています。つまり

、APIではイテレータの使用が許可されていません。

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、次の要素を持つ基本XMLクエリ以外の異なるヘッダーを使用できません。

.
.

.

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あります。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：

DEFAULT

VRF_NAME

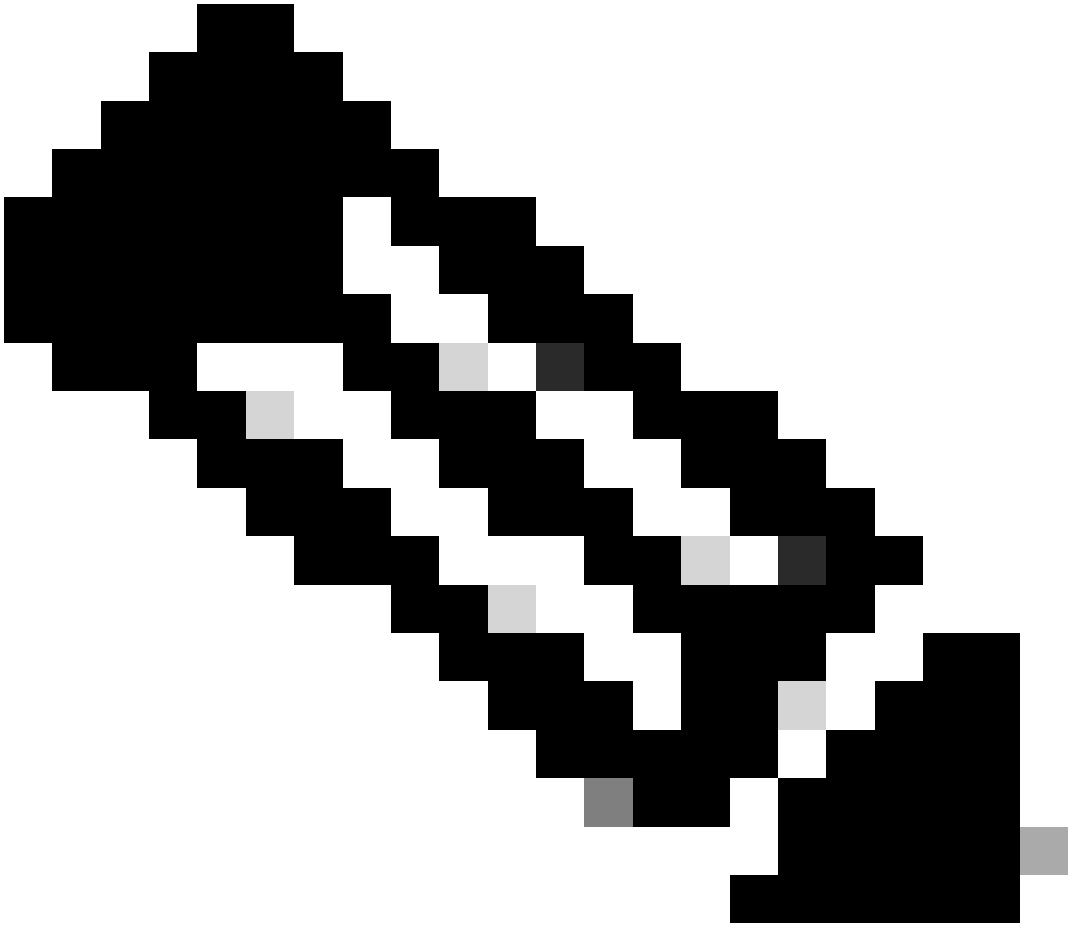
NAME

>

このクエリはフルBGPテーブルを対象とします。この非固有の要求に対する応答はほぼ2.2 Gbでした。したがって、メモリスロットルの制限に達しています。

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、データの表示および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理解するものを解析し、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。

```
This is an example
```

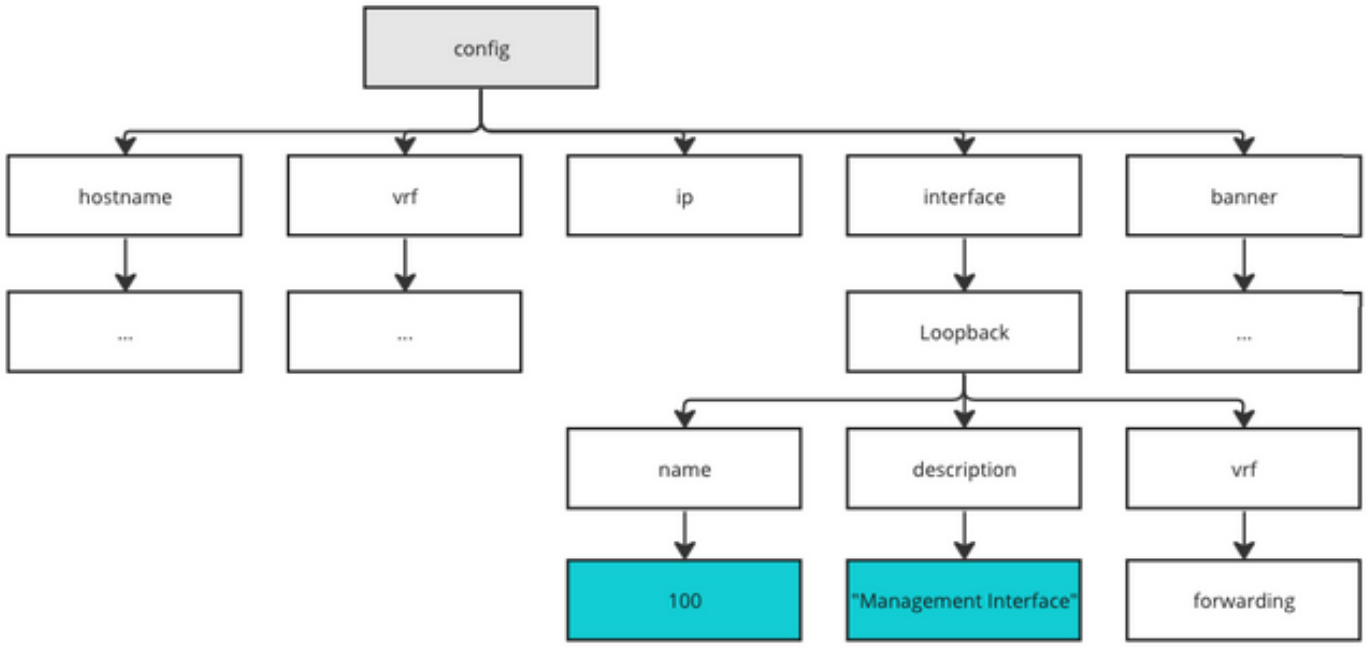
すべてのタグには、開始タグ(<init>)と終了タグ(</init>)の2つの部分があります。 この構造が使用されていない場合、XMLはタグが終わる場所を理解できません。

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、すべてのブランチにはスタブされたリーフがあります。リーフには情報以外は含まれません。

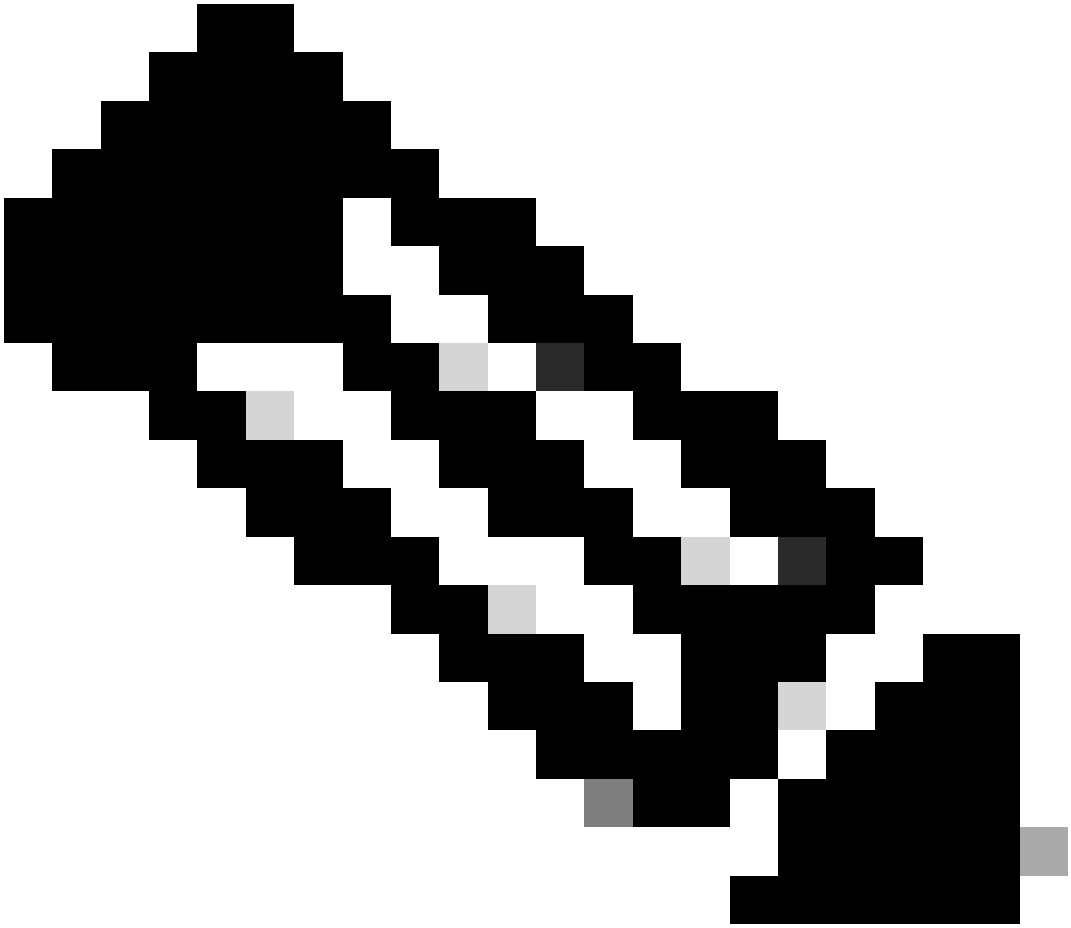
たとえば、次の木のルートは灰色、コンテナは白、リーフは青です。



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. クエリにヘッダーがある場合：



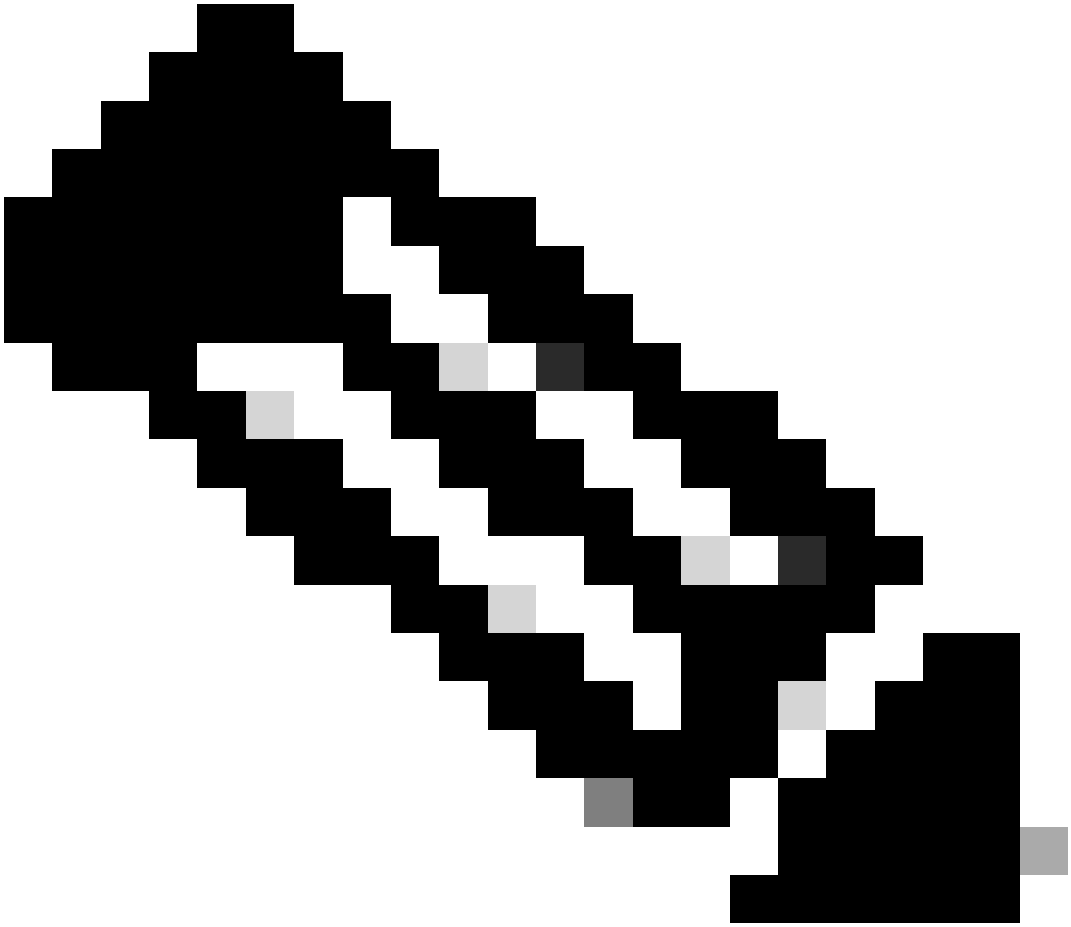
注：これは、終了タグを必要としない唯一のタグです。

---

2. 要求に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。

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。

...



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、必要な情報をすべて含む要求を表示します。

…

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：

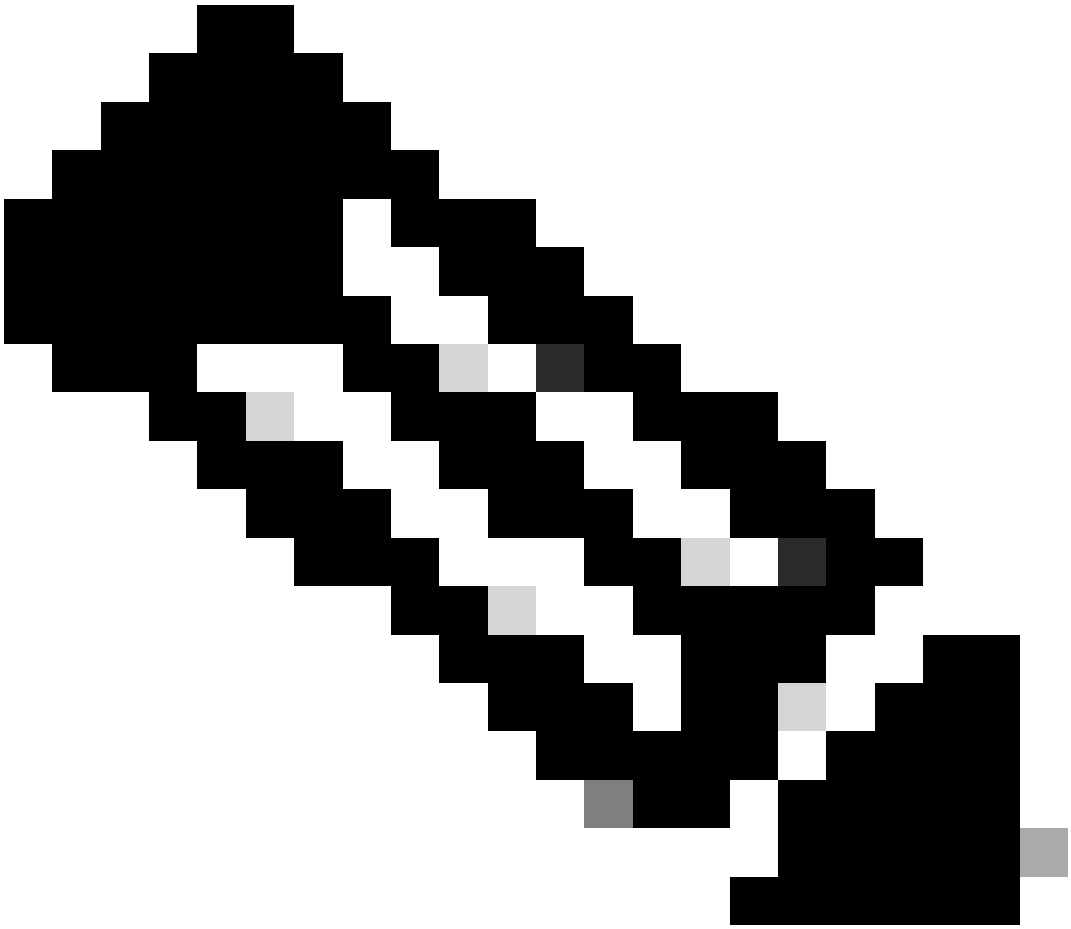
- <Get></Get>：アクションデータ項目（リーフに含まれるデータ）を収集します。
- <セット></セット>：データ項目を構成、変更、または追加します。
- <Delete></Delete>:1つ以上のデータ項目を削除します。
- <GetVersionInfo></GetVersionInfo>：要求されたコンポーネントのメジャーバージョン番号とマイナーバージョン番号を取得する操作。
- <GetDataSpaceInfo></GetDataSpaceInfo>：この操作タグは、コンテナにマッピングされたリーフ名を表示します。

2. CLI操作：

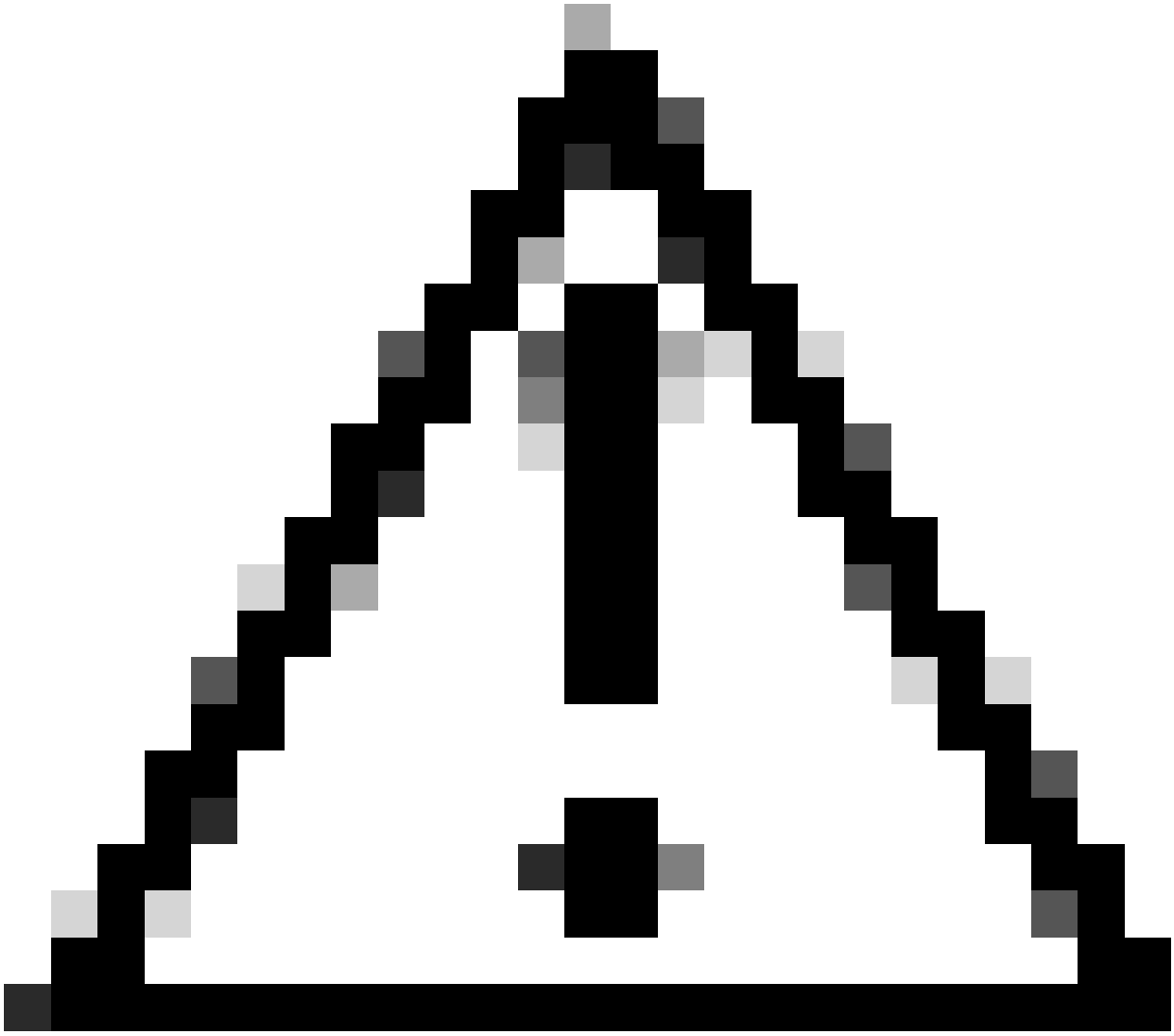
- <CLI><CLI/>:CLIコマンドとして設定要求を発行できるタグ。
- <GetNext><GetNext/>：データがブロックサイズより大きい場合、このタグは次のセグメント化された応答に追加されます。

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こと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：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：管理運用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。

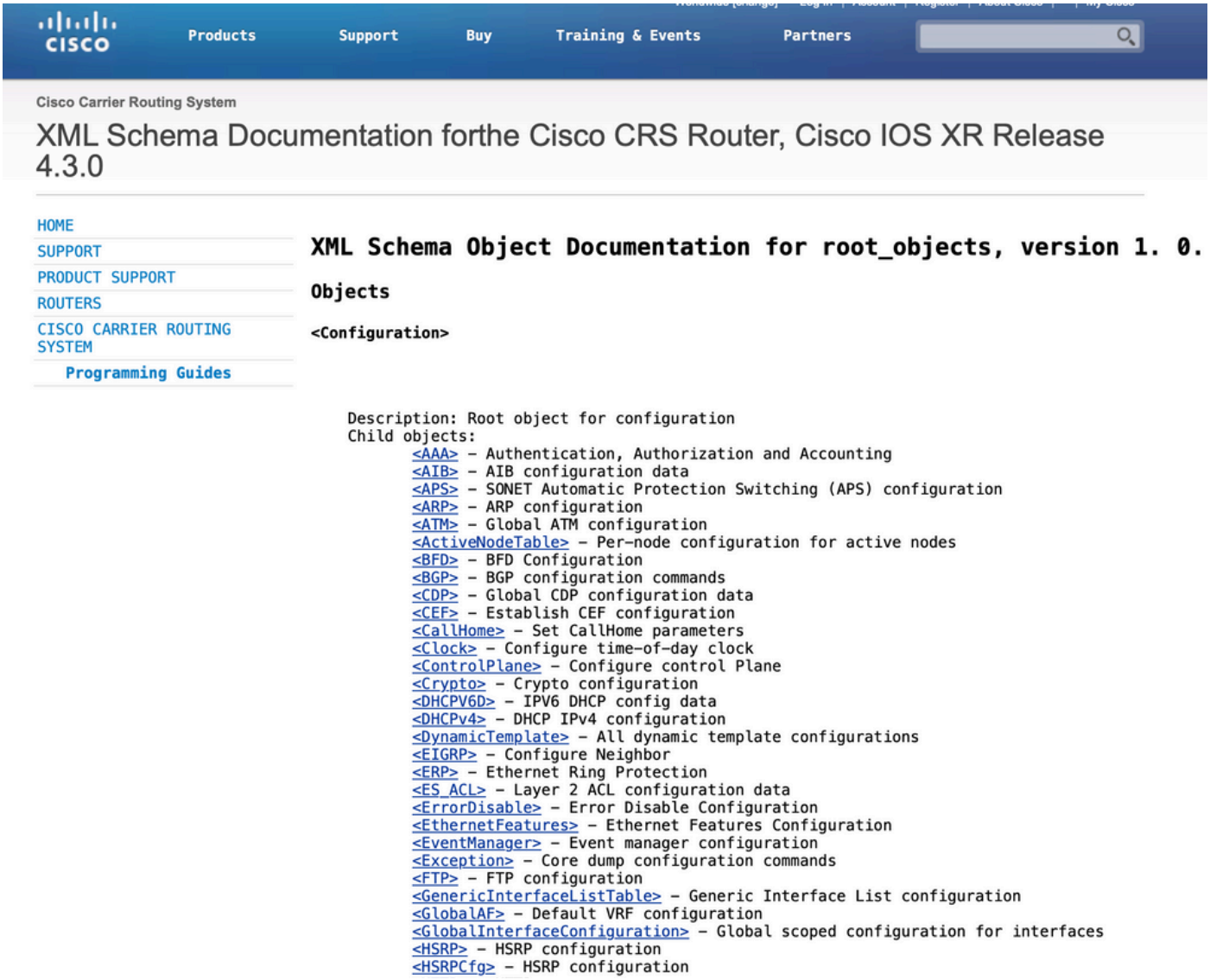
クエリーの作成

移動には2つの異なる経路があります。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スキーマ文書

これらのオプションは、コンテナとリーフを示す完全な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。



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。

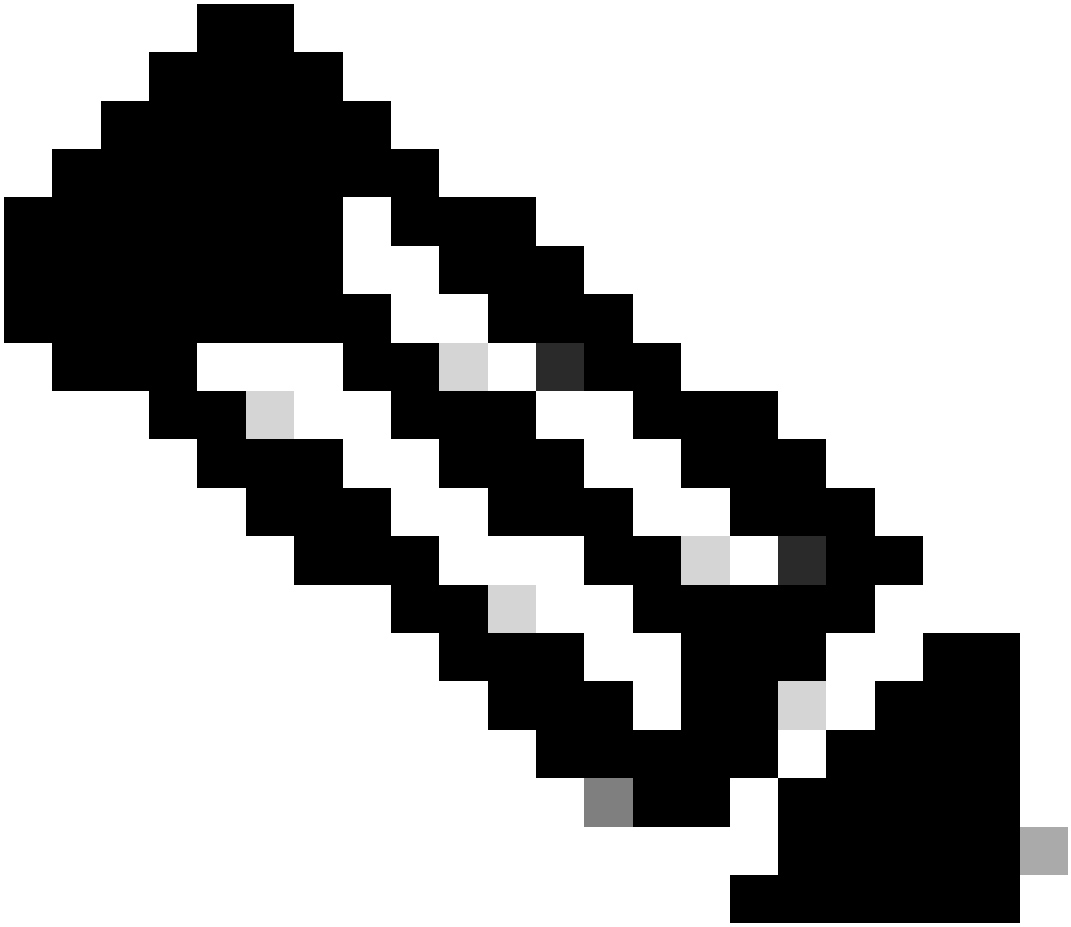
子をクリックすると、クエリーを作成するための情報が表示されます。IPv4の例：

```
<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、そして最後にリーフIPv4を対象とします。



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。

2. XML API

XML APIはCisco IOS XRI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。このコマンドを実行すると、Linuxと同様に、XMLスキーマ全体がCLIに表示されます。

このCLIでは、次のアクションがあります。

?: – ヘルプと同じ

LS:ブランチユーザのコンテナおよびリーフをリストします。

<operation>:?を発行すると、操作を確認でき、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
```

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-ls/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
```

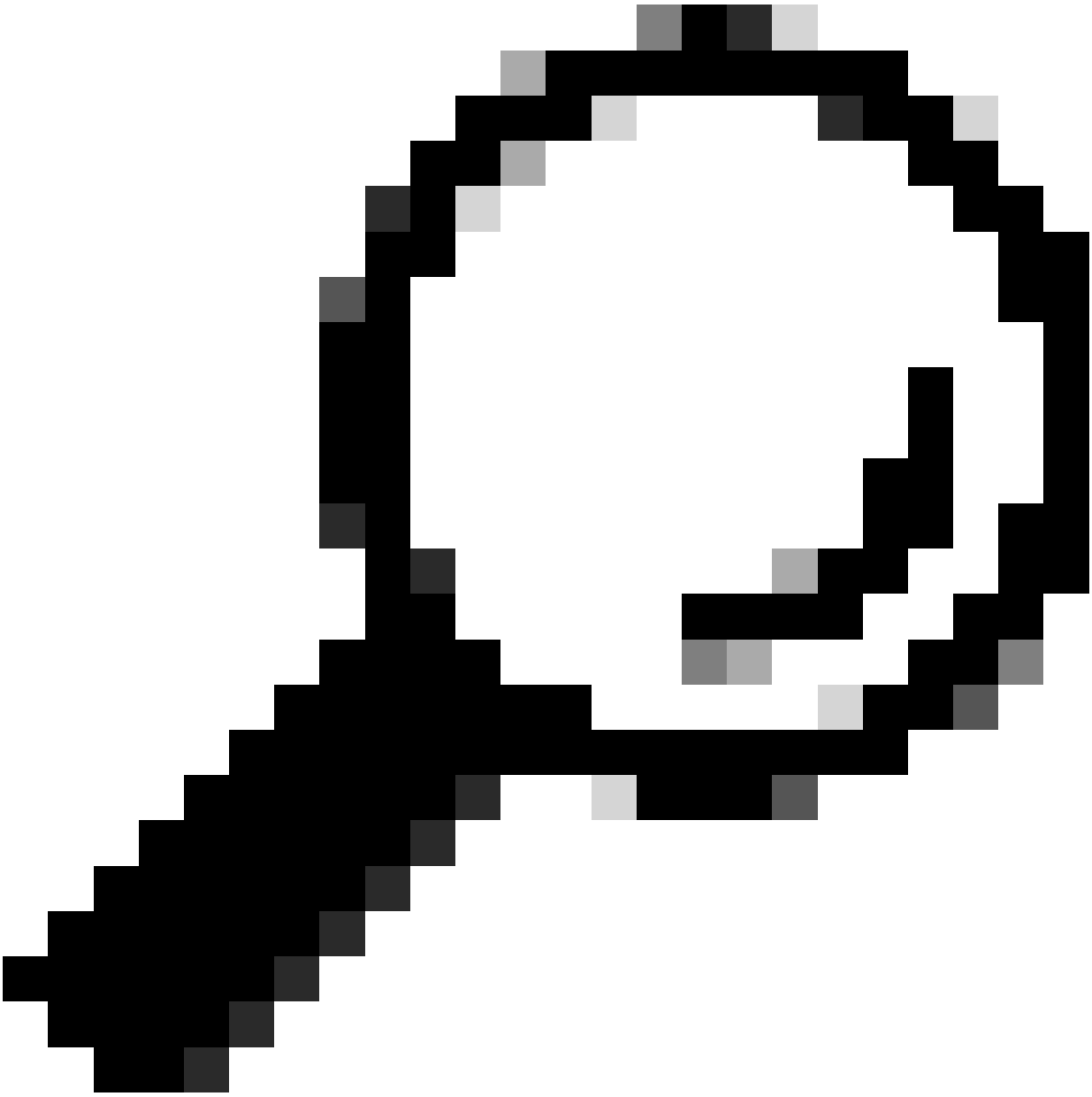
次のプロンプトに注目してください。

```
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、クエリは次のようになります。

XMLクエリーのテスト

クエリーが作成されたら、次の手順でそれをテストします。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：前述のコマンドを入力した後で、ユーザがクエリをコピー/貼り付けしてEnterキーを押せ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>
```

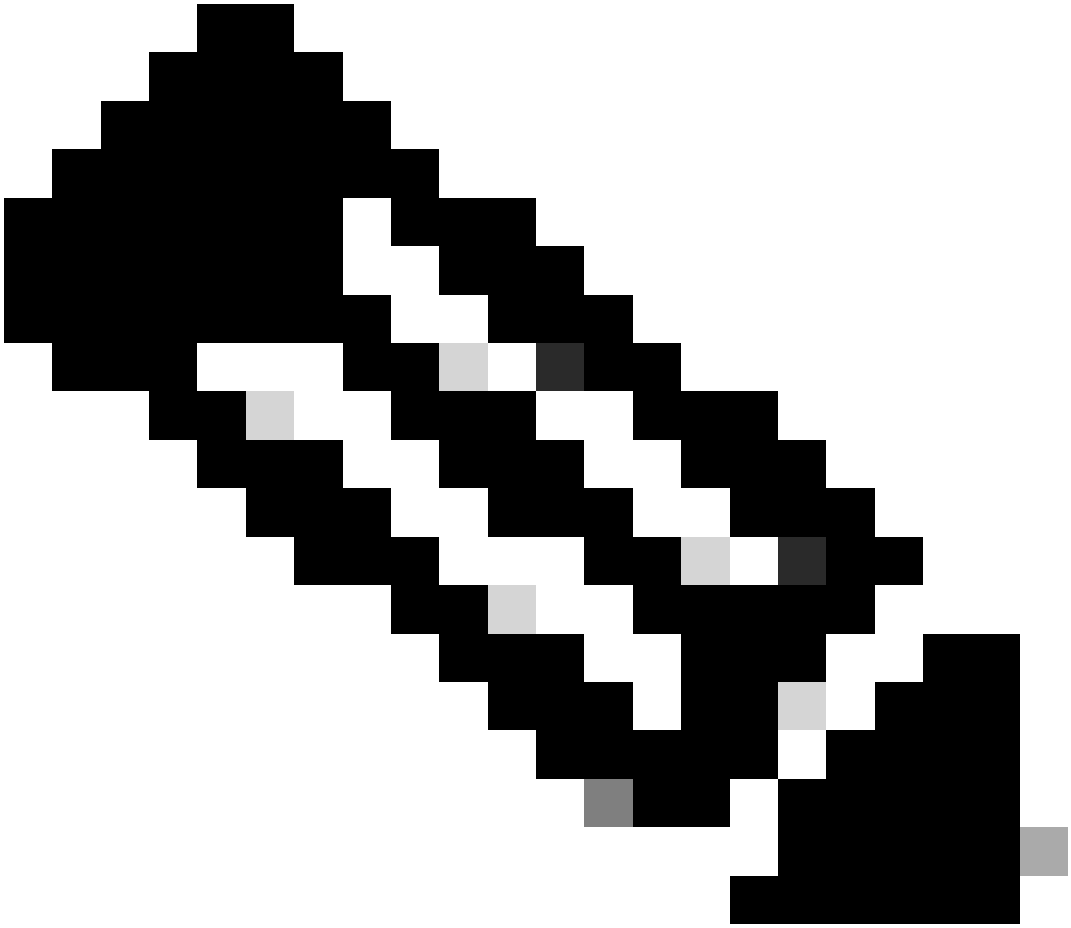
>>> Hit enter here. Immediately getting the reply.

! Reply:

true >>> This is the requested value.

```
XML> exit
```

ルータは応答とResultSummaryを表示します。



注：このXML応答は、スロットルメモリを表示せず、設定する必要もありません。

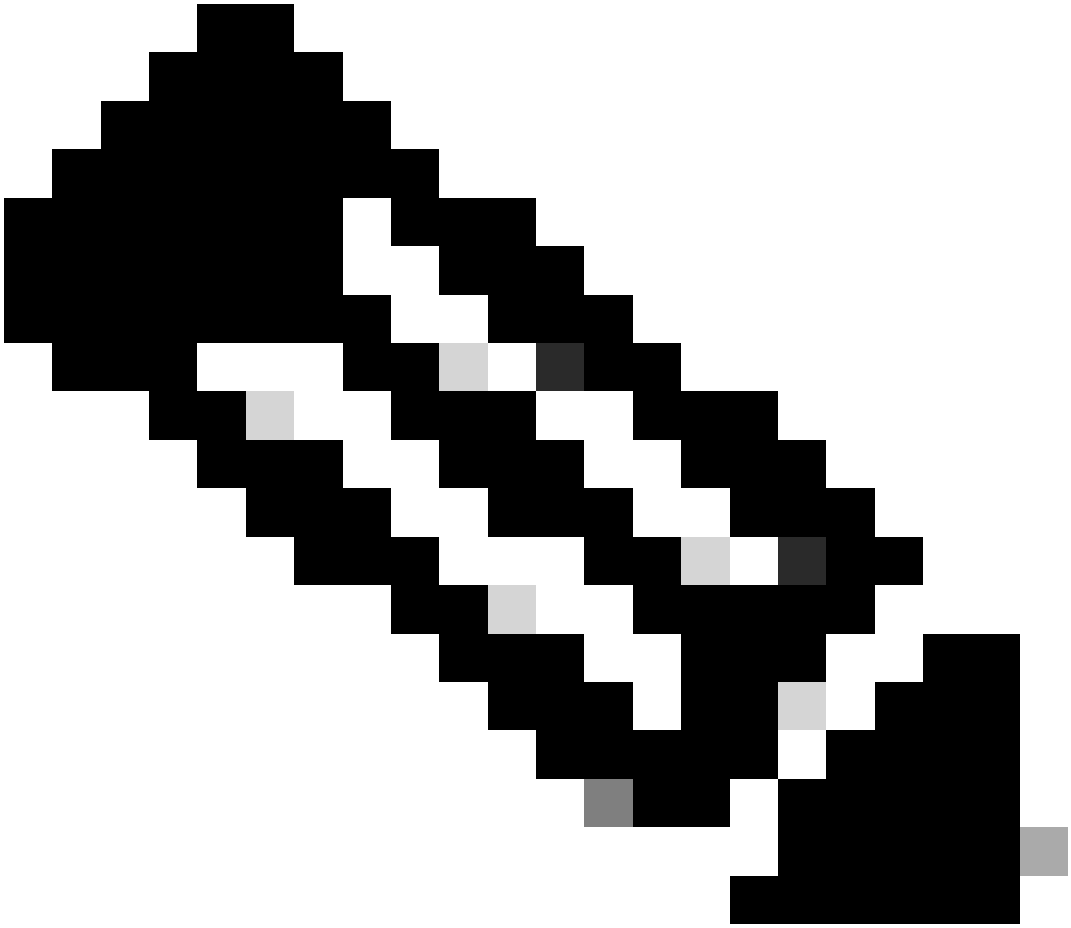
SSH/TTYを使用していないため、サイズに関係なく応答が要求されます。1つはXMLクライアントで、もう1つは埋め込みXML echoコマンドです。動作は変更される可能性があります。

Cisco IOS XRでのXMLのディープダイブ

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が開始されたとき、デバイスでXMLを利用するには2つの異なる方法がありました。

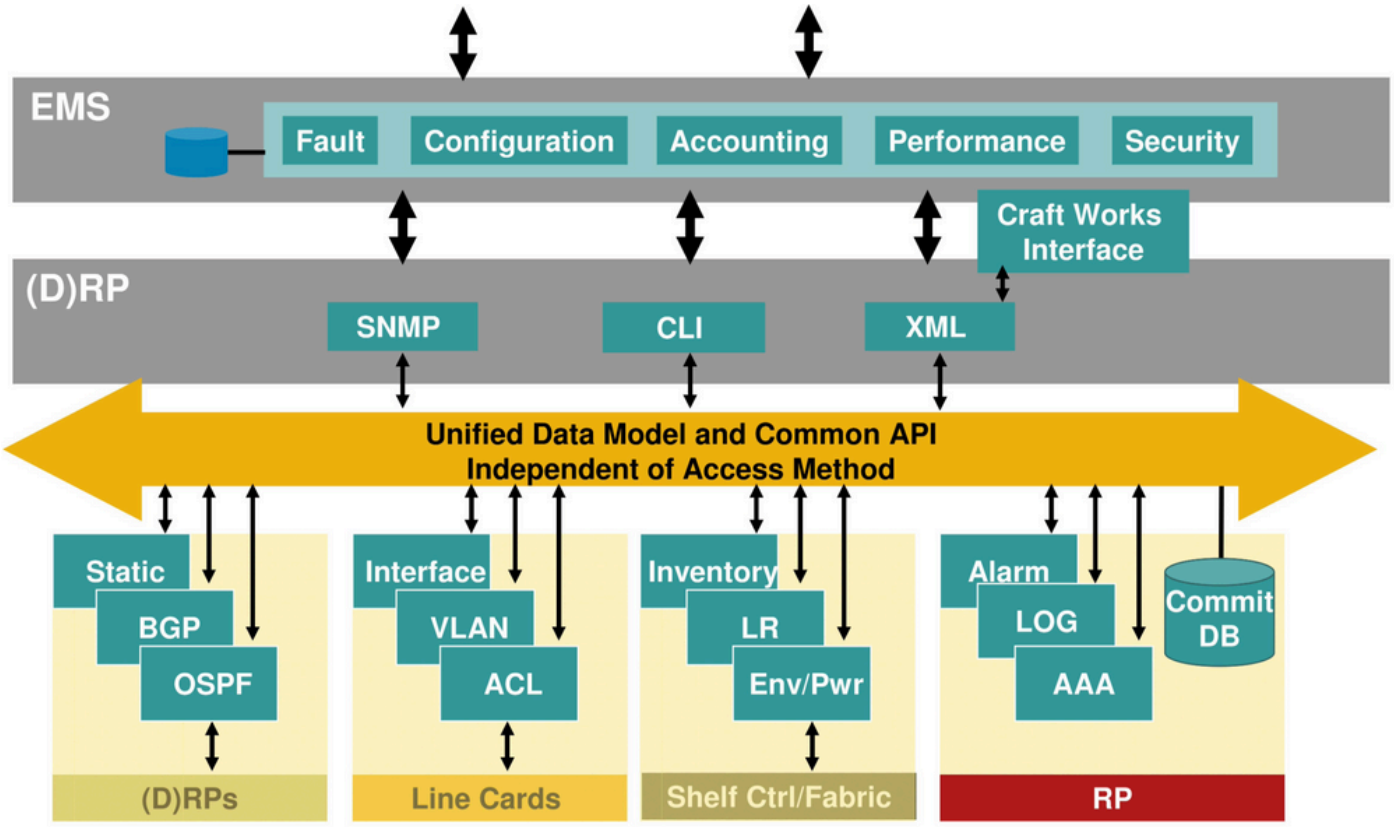
1. CORBAエージェント（SSLを使用）
2. SSHエージェント。



注:CORBAはバージョン3.7以降で廃止されています。

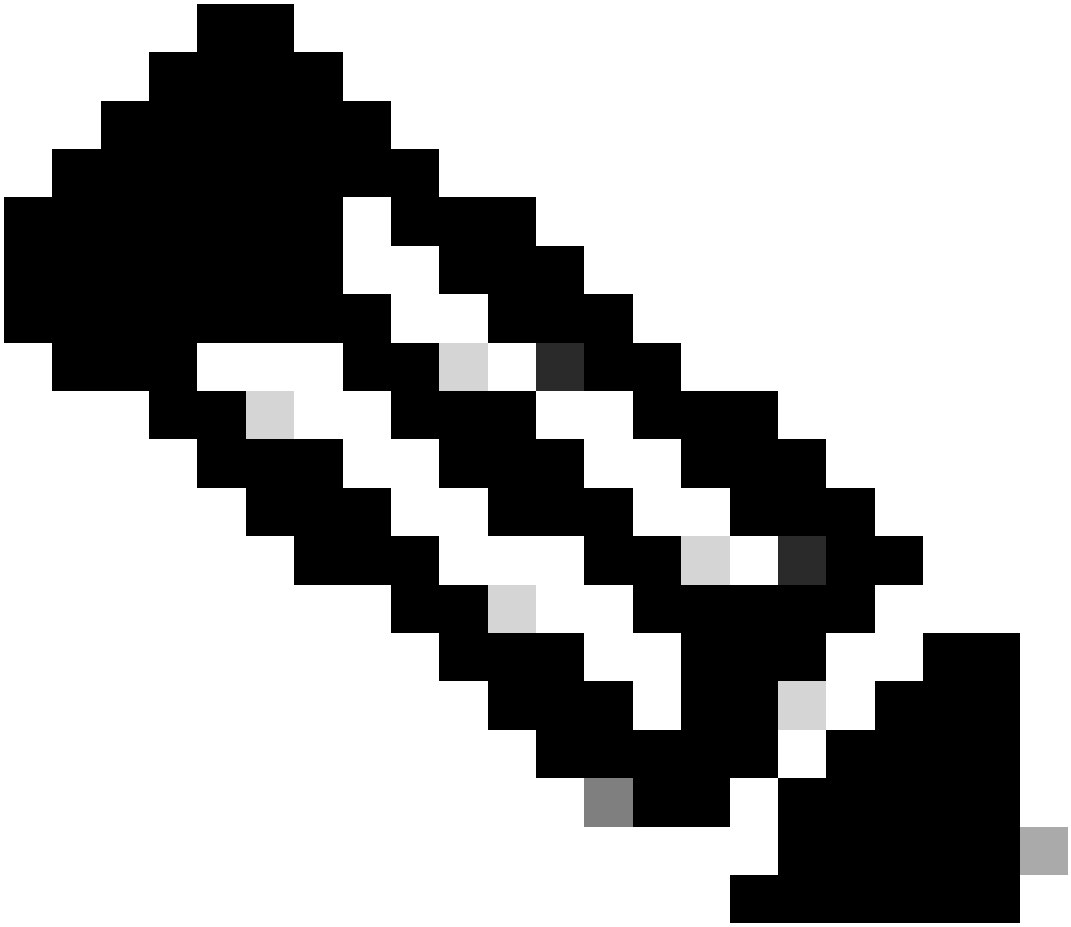
CORBAは、従来のプロトコルとしてSSLを使用しており、これはセキュリティパッケージに含まれています。一方、SSH方式では、マネージャビリティー・パッケージを使用する必要があります。

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情報をすべて監視する共通のAPIが含まれています。これらの情報は次のとおりです。

- 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ではTCP 38751を使用)
  - Netio

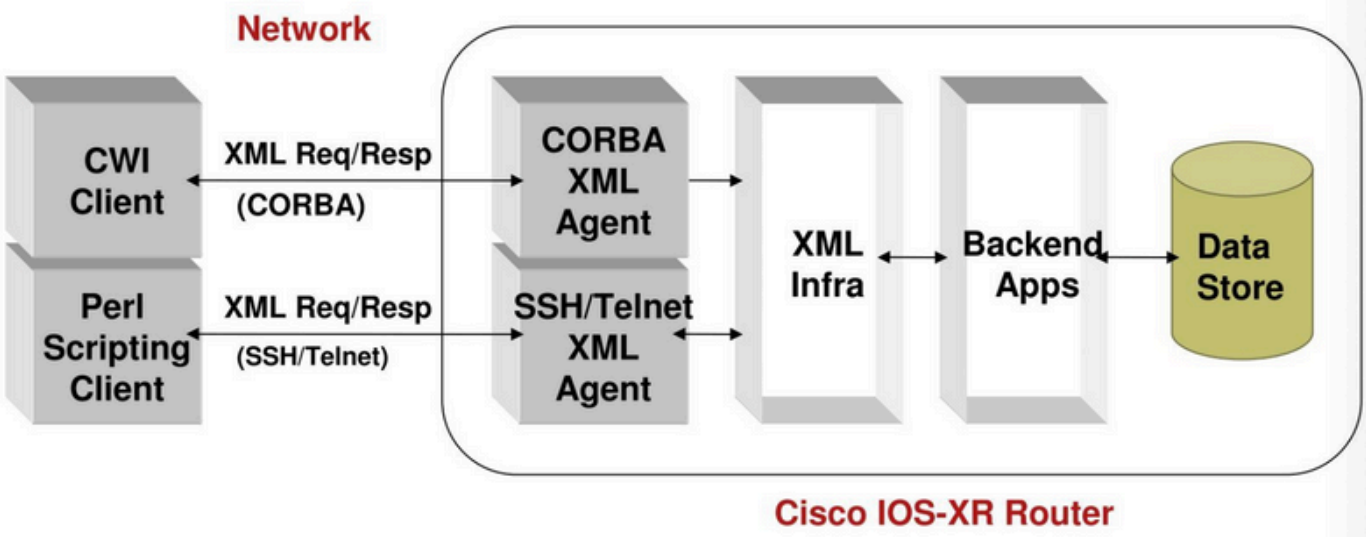


注：これに関する詳細は、「Cisco IOS XRのXML工ラー」セクションを確認してください。

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。

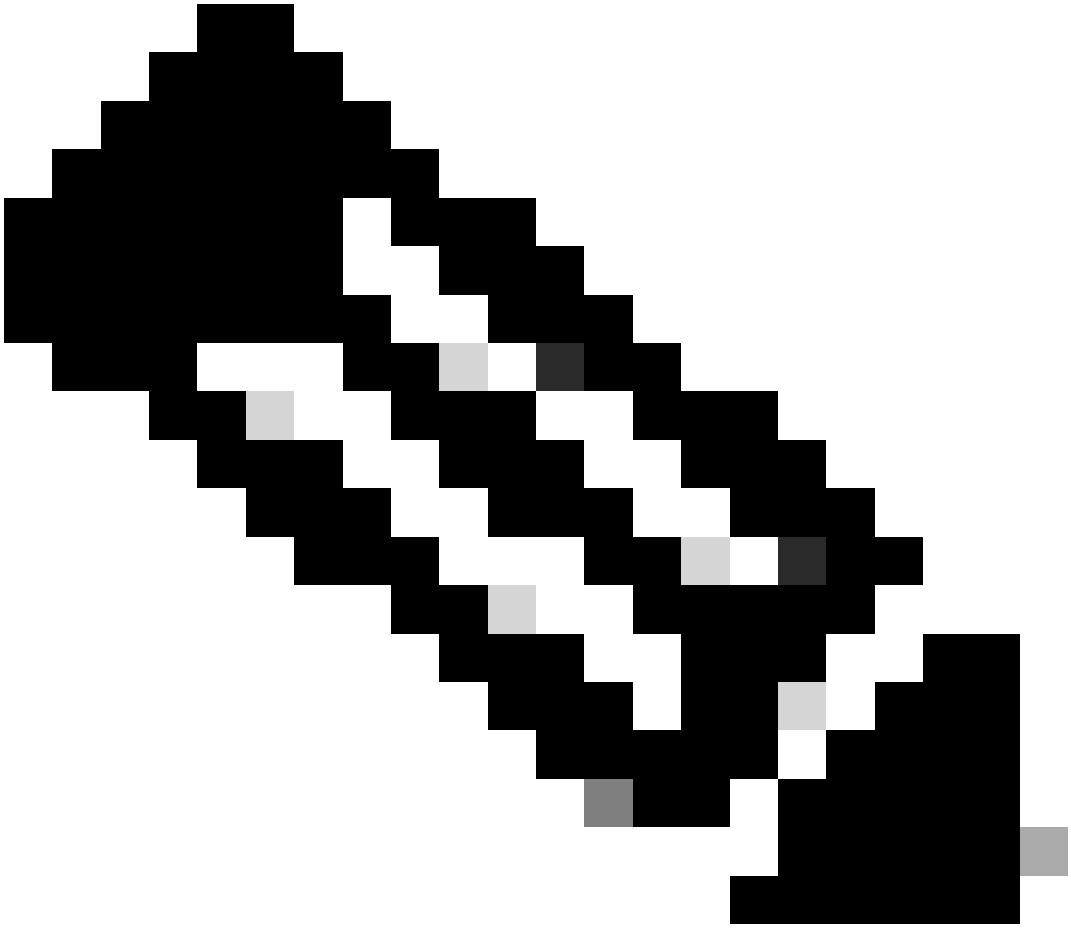
1. バージョン
2. スキーマ

バージョンは、次の2つの領域で定義できます。

- 要求タグ：

- 主コンポーネントタグ：

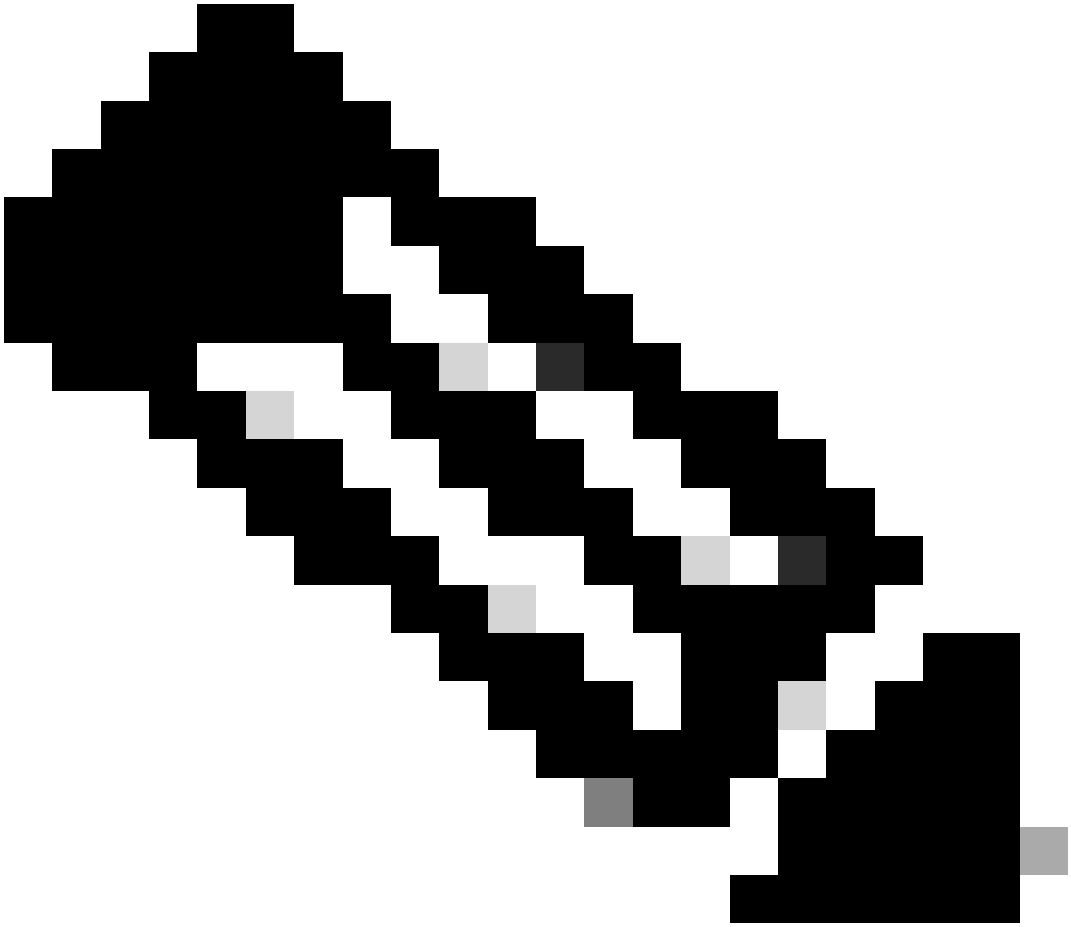
これは、特定のコンポーネントに適用されます。



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：新規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。



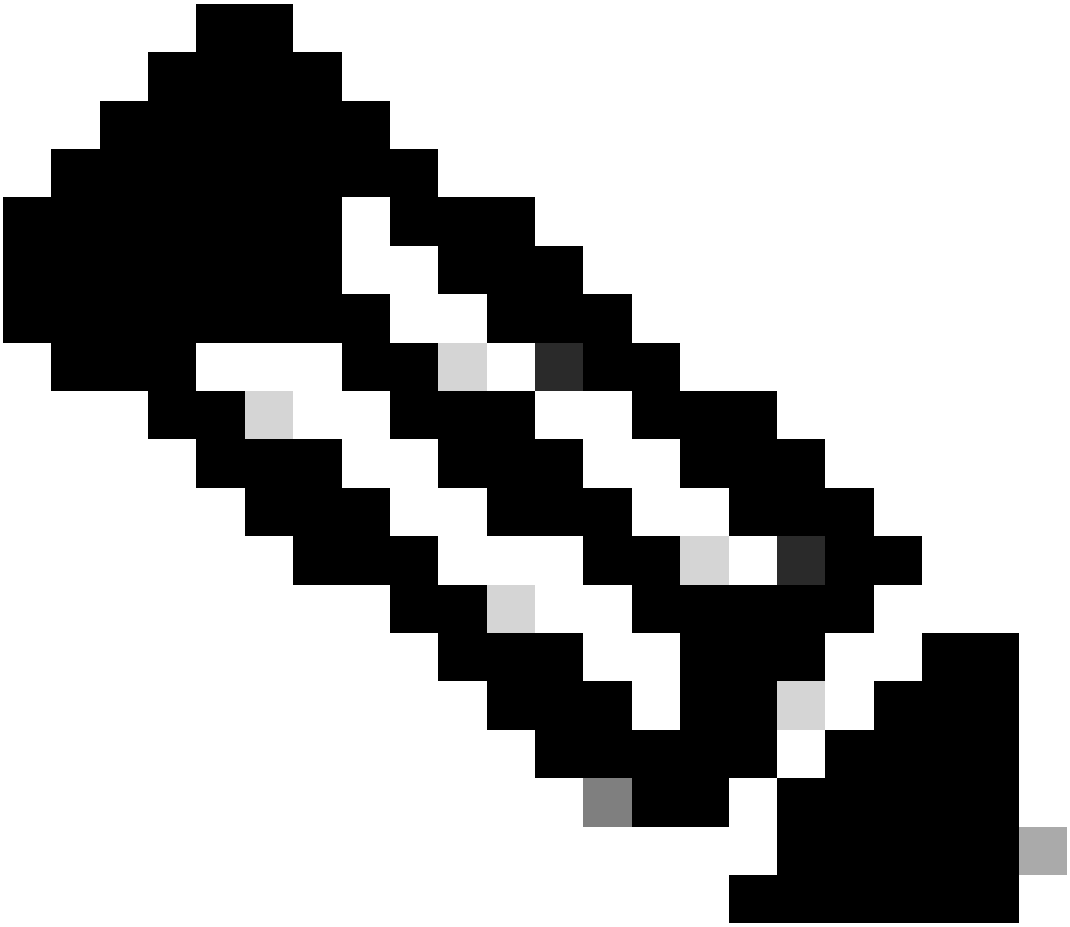
注:XML操作は、GET、SETなどです。

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検索します。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>
```

```
        >>> This is the request
```

XML>



注意：要求を実行すると、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、その中のコンテナ内のバージョンも表示されます。

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。

クエリが正しくない場合

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。

- 項目未検出（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. トランスポート：このカテゴリのエラーには、XMLエージェント/クライアント通信の間にあるものがすべて含まれます。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、これらの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、送信された応答またはクエリに問題がある。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：

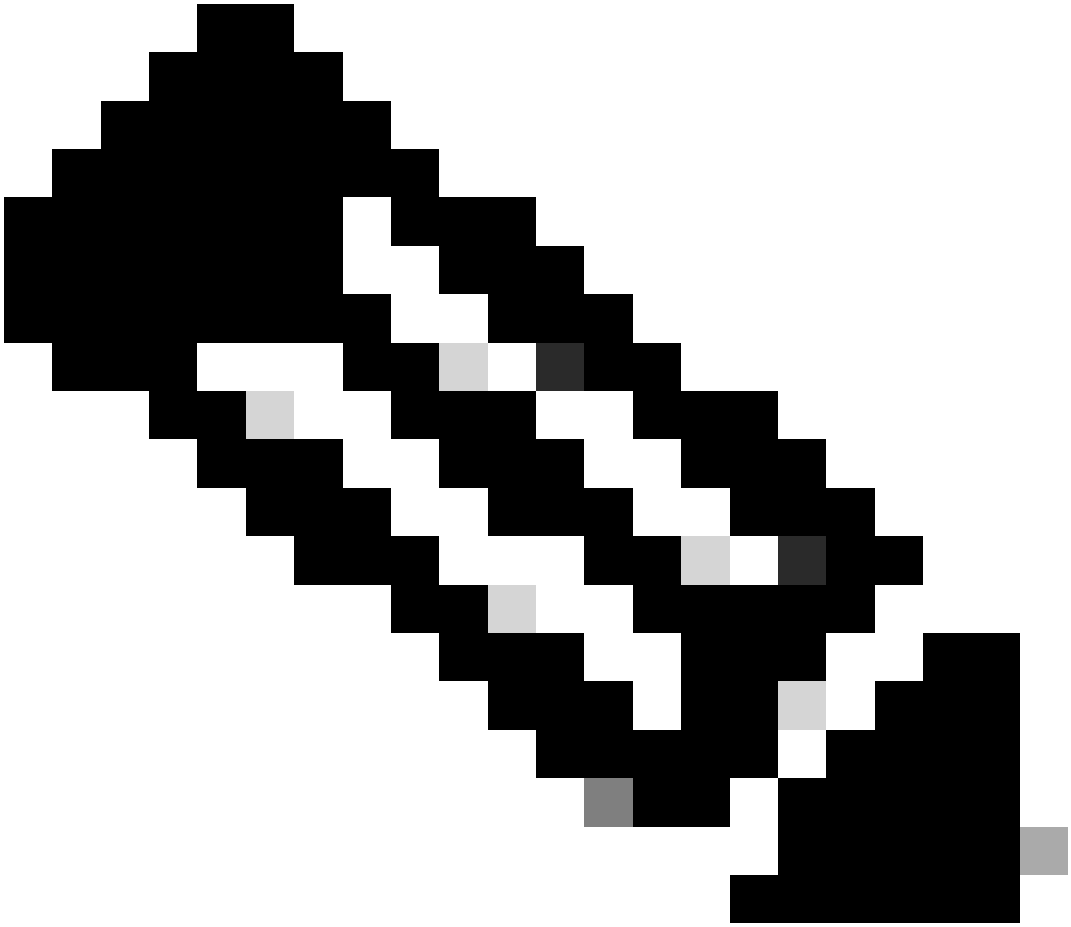
```
ERROR: 0xa367a600 'XML Service Library' detected the 'fatal' condition 'The throttle on the memory usag
```

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" ErrorMsg="'XMLMDA' detected the 'warning' condition 'An XML request contains an
```

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ビットのint）属性とErrormsg属性の形式でコード化されています。

その他の推奨事項

その他の有用なテクニック：

1. ワイルドカード：これは特定クエリとも呼ばれます。
2. バッチ処理：複数の手法や操作を1回のリクエストに組み合わせる（ベストエフォート操作）。
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。