

# Identificar e Solucionar Problemas do XML para Cisco IOS XR

## Contents

---

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Acelerar problemas de memória](#)

[Descrição do problema adicional](#)

[Solução](#)

[XML no Cisco IOS XR](#)

[Usando API XML](#)

[Operações com Suporte da API XML](#)

[Serviços oferecidos do esquema XML](#)

[Criando uma Consulta](#)

[Testando Consultas XML](#)

[Mergulho profundo em XML no Cisco IOS XR](#)

[Interação de API com cliente](#)

[Erros XML no Cisco IOS XR](#)

[Quando a consulta está incorreta](#)

[Tipos de erro](#)

[Outras recomendações](#)

---

## Introdução

Este documento descreve problemas de aceleração de memória XML vistos em solicitações de serviço e ferramentas da Cisco.

## Pré-requisitos

### Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Plataforma Cisco IOS® XR
- Infraestrutura XML (Extensible Markup Language)
- Arquitetura Common Object Request Broker (CORBA)
- Sistema de gerenciamento de elementos (EMS)
- Gerenciador de dados externos (EDM)

## Componentes Utilizados

As informações neste documento são baseadas no ASR9000.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

## Acelerar problemas de memória

Um caso foi aberto com estas perguntas:

1. Qual é a memória máxima configurável no roteador?
2. Há uma maneira de quebrar respostas XML grandes?

Resposta:

1. A alteração de memória máxima depende da versão que o RSP/nó está executando (versões para cXR e eXR)

Na versão de 64 bits (eXR). Você tem um tamanho de memória de aceleração:

```
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)
```

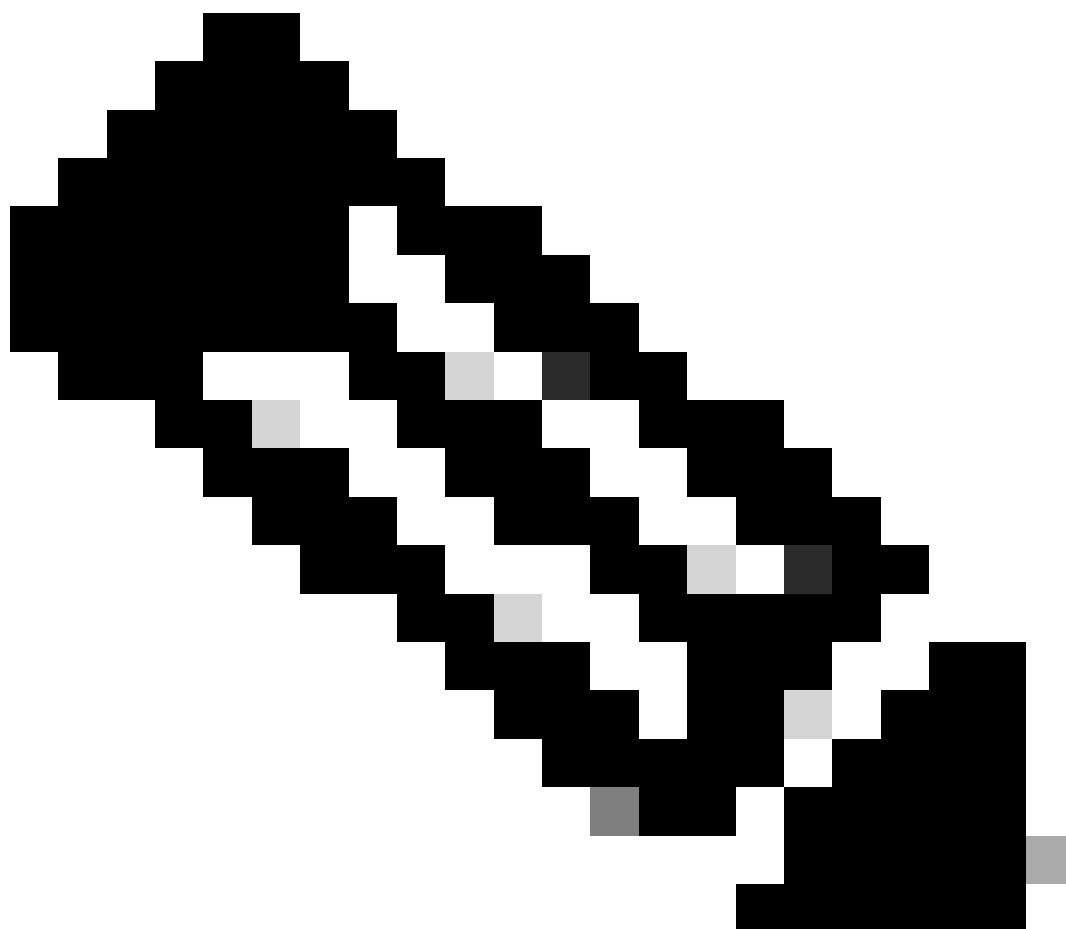
Na versão de 32 bits (cXR). Você tem outro limite:

```
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)
```



Observação: o default em qualquer versão é 300 Mbytes

---

2. Há diferentes maneiras de quebrar a resposta XML:

- Iteradores:

Quando os Iteradores são configurados, você segmenta a resposta total XML em um tamanho de janela específico. A janela tem o tamanho do Iterador. Por exemplo, se a resposta for 1 Gb e o tamanho do Iterador for 500 Mb, segmente o XML em 2 respostas.

Esse método altera a resposta adicionando (com base no tamanho do Iterador) operações GetNext (o que a resposta precisa ).

Pode haver até 10 iteradores para uma sessão.

- Limitação (taxa de processo):

Esse recurso limita a memória consumida pelo processo XML. No caso de um processo ultrapassar a memória, ele responde com a mensagem de erro: "O acelerador do uso de memória foi atingido".

- Transmissão:

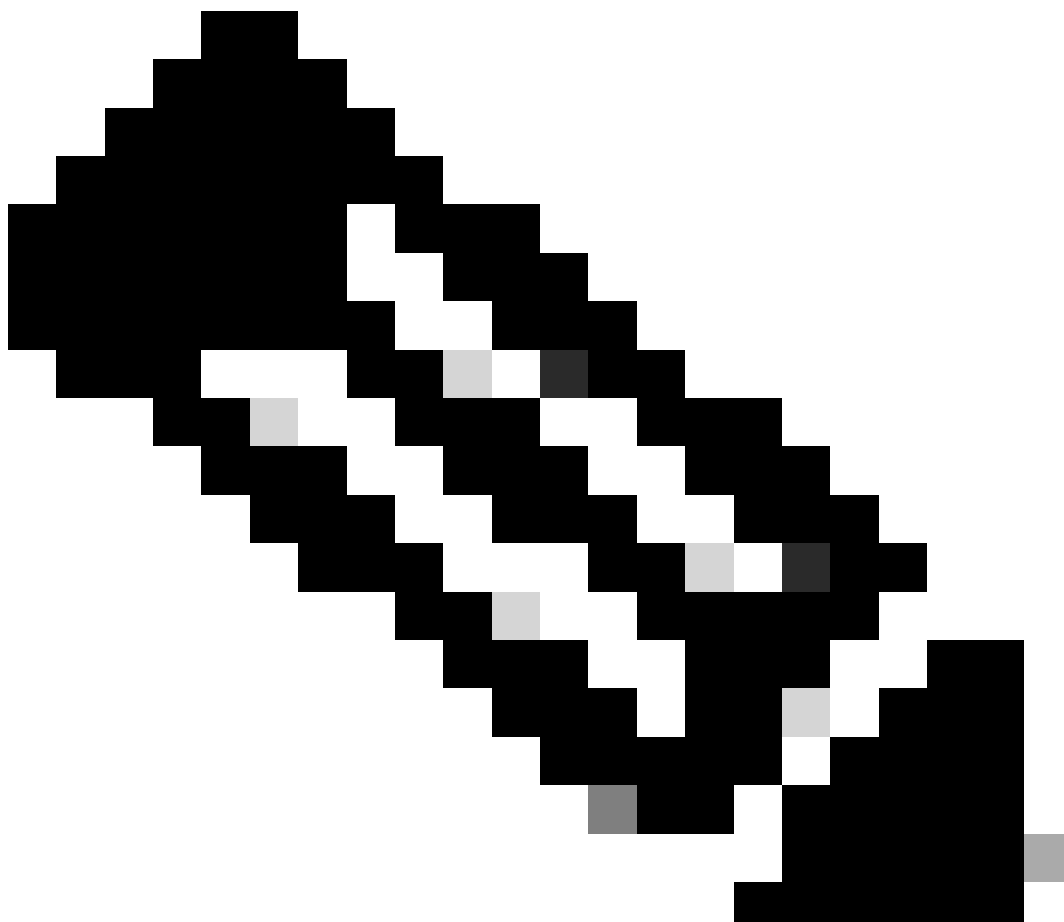
Semelhante aos Iteradores, o fluxo segmenta a resposta XML em janelas específicas. A diferença é a resposta, ela remove a operação GetNext e a ID do Iterador. O XML envia os pacotes transmitidos e o cliente constrói a resposta quando o fluxo termina.

## Descrição do problema adicional

Para casos de uso em que a automação é necessária, use a ferramenta [pyIOSXR](#). Essa ferramenta de automação é um agente XML, ajuda a emitir alguns comandos show e, em geral, se conecta ao dispositivo.

Toda vez que você enviar uma solicitação grande com este agente, um erro será exibido:

---



Observação: somente usuários registrados da Cisco podem acessar ferramentas e informações internas da Cisco.

---

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

Com a mensagem de erro exibindo "otimizar a solicitação para consultar dados menores", você pode configurar a API XML do agente com Iteradores. Isso, em teoria, permite segmentar a resposta.

Quando os iteradores foram configurados, a mesma mensagem de erro foi exibida: "ERRO DE RESPOSTA...Otimize a solicitação para consultar dados menores".

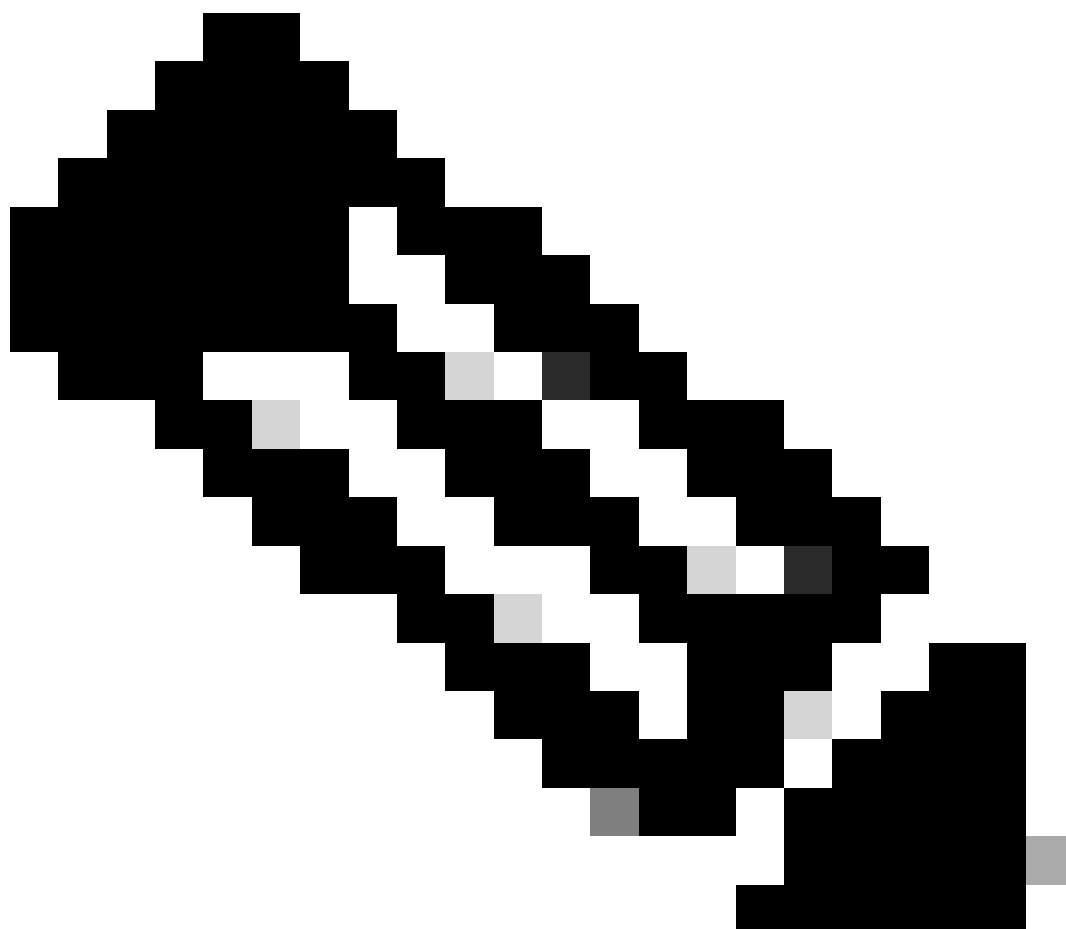
Quando um erro é exibido, a próxima etapa é entender por que o recurso, neste caso Iteradores, não funcionou para a consulta.

O [pyIOSXR](#) recomenda que o agente XML seja habilitado corretamente no dispositivo, o que

significa que a API não permite o uso de Iteradores.

A próxima etapa é testar a segunda opção: Transmissão.

---



Observação: pyIOSXR não nos permite trabalhar com cabeçalhos diferentes dos da consulta XML básica, que tem os próximos elementos:

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

---

O fluxo e os iteradores adicionam um cabeçalho nas consultas. O fluxo adiciona um ID de fluxo

que ajuda o sistema a criar a resposta completa. Iteradores adicionam GET\_Next e Iterator ID.

O teste de streaming também mostrou a mesma mensagem de erro dos Iteradores.

## Solução

Na seção, Mais recomendações para fazer consultas ao dispositivo, há mais ferramentas para resolver esses problemas. O curinga é um deles. Curinga é a solução para o limite de aceleração de memória.

O curinga cria uma consulta específica para evitar a solicitação de informações desnecessárias. Por exemplo, para informações de BGP, use o comando show route bgp em vez do comando genérico show route. Este exemplo se aplica às consultas e à lógica XML. A solicitação de informações em massa ao sistema pode gerar problemas de memória e processamento.

Quando o caso foi aberto, esta consulta foi usada:

```
<?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>
```

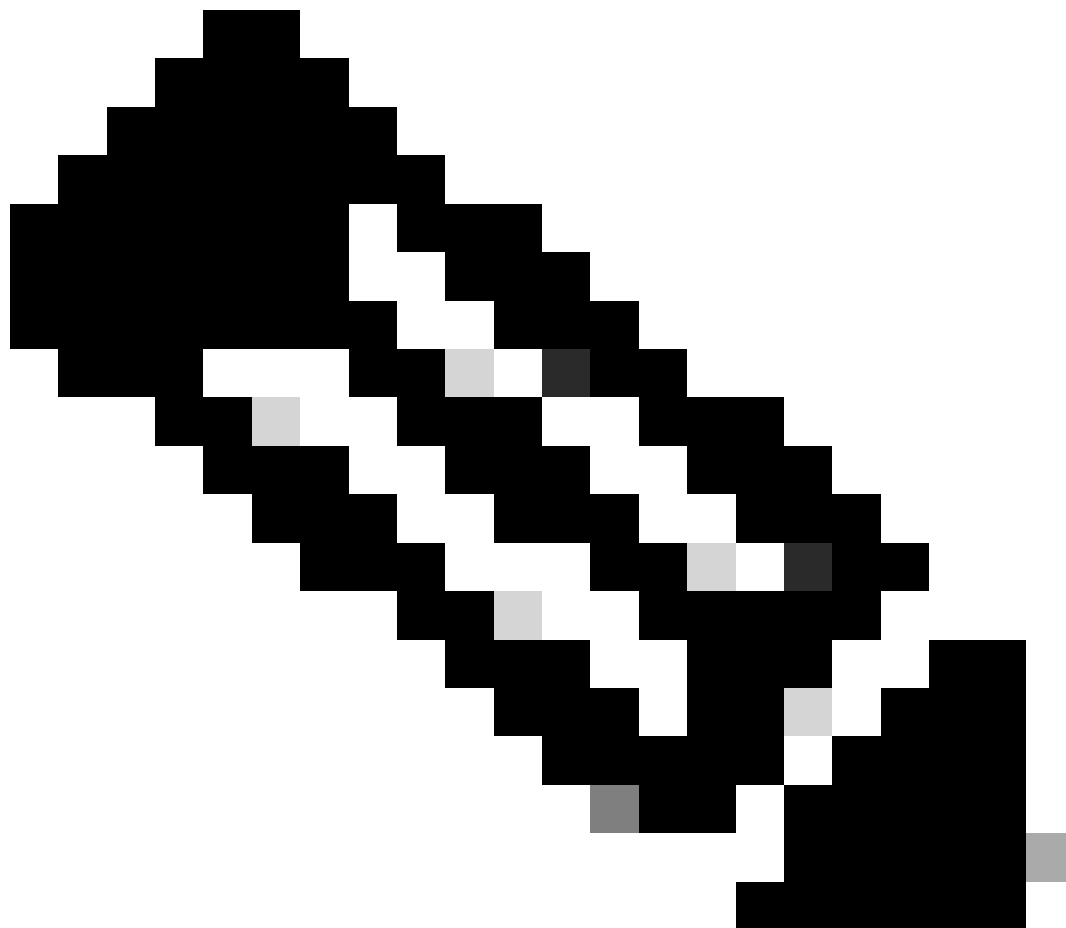
```
</Operational>  
</Get>  
</Request>
```

Esta consulta é direcionada para as tabelas BGP completas. Para essa solicitação não específica, a resposta foi de quase 2,2 Gb, portanto, o limite de aceleração de memória é atingido.

Para corrigi-la, é necessária uma consulta específica, o que permite que o sistema processe a consulta e retorne as informações.

## XML no Cisco IOS XR

---



Observação: caso sejam necessárias mais informações, especificações ou outros comandos, clique no próximo link Guia de programação XML: [Guia de programação XML XR](#)

---



O XML define como os dados são exibidos e estruturados. Essa é uma maneira de analisar o que o computador entende como bits e exibir informações estruturadas e padronizadas.

O XML tem esta estrutura:

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

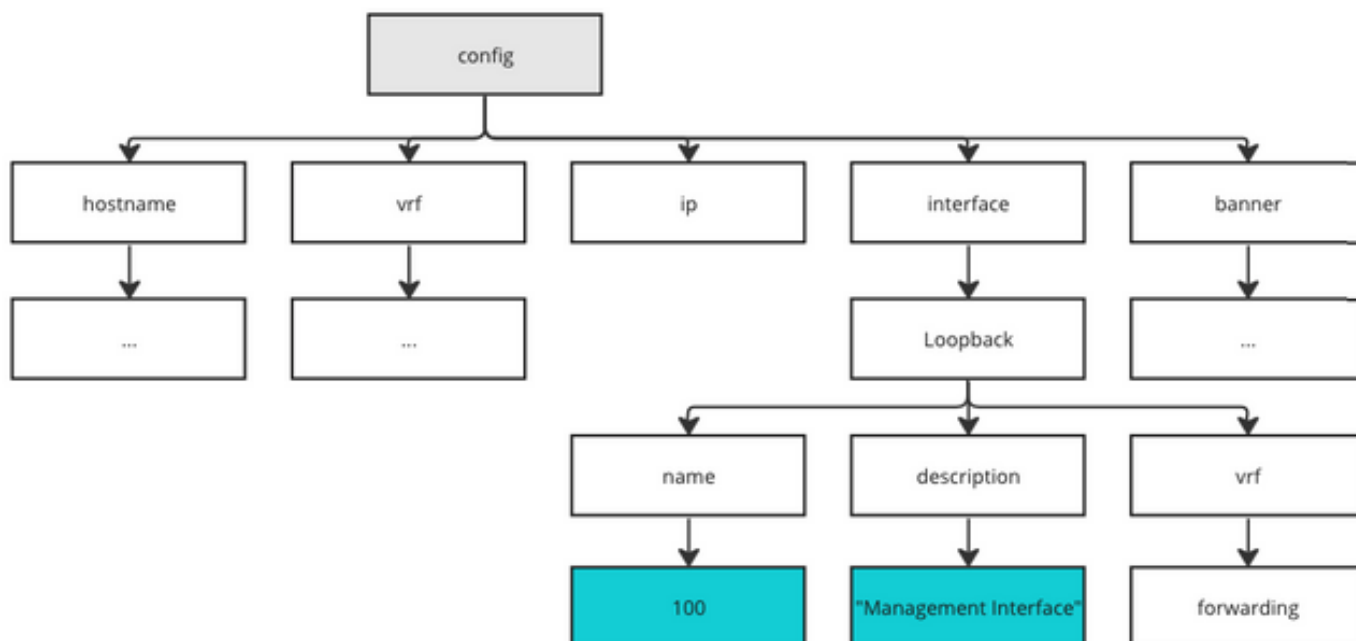
Todas as tags têm duas partes, a tag de abertura (<init>) e a tag de fechamento (</init>). Se essa estrutura não for usada, o XML não poderá entender onde uma marca de formatação termina.

O XML é composto de duas entidades:

- Contêineres
- Folhas

Uma analogia para esses dados estruturados é comparar os dados XML como uma árvore. Um recipiente é um ramo e cada ramo tem folhas esbugalhadas. Os folhetos não contêm qualquer outra informação.

Por exemplo, a próxima árvore tem a raiz em cinza, contêineres em branco e folhas em azul.



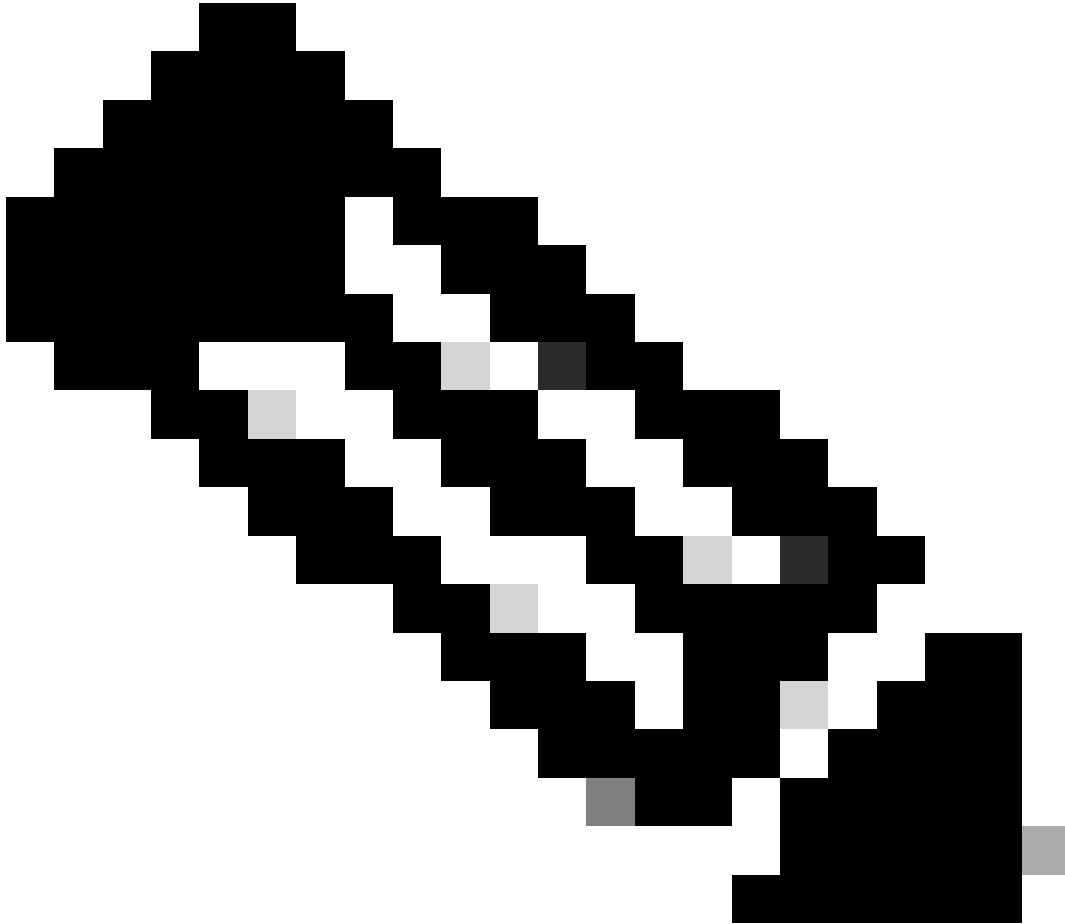
## Usando API XML

Para testar/utilizar a API XML, a primeira coisa de que você precisa é uma consulta.

1. A consulta tem um cabeçalho:

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

---



Observação: esta é a única tag que não precisa de uma tag de fechamento.

---

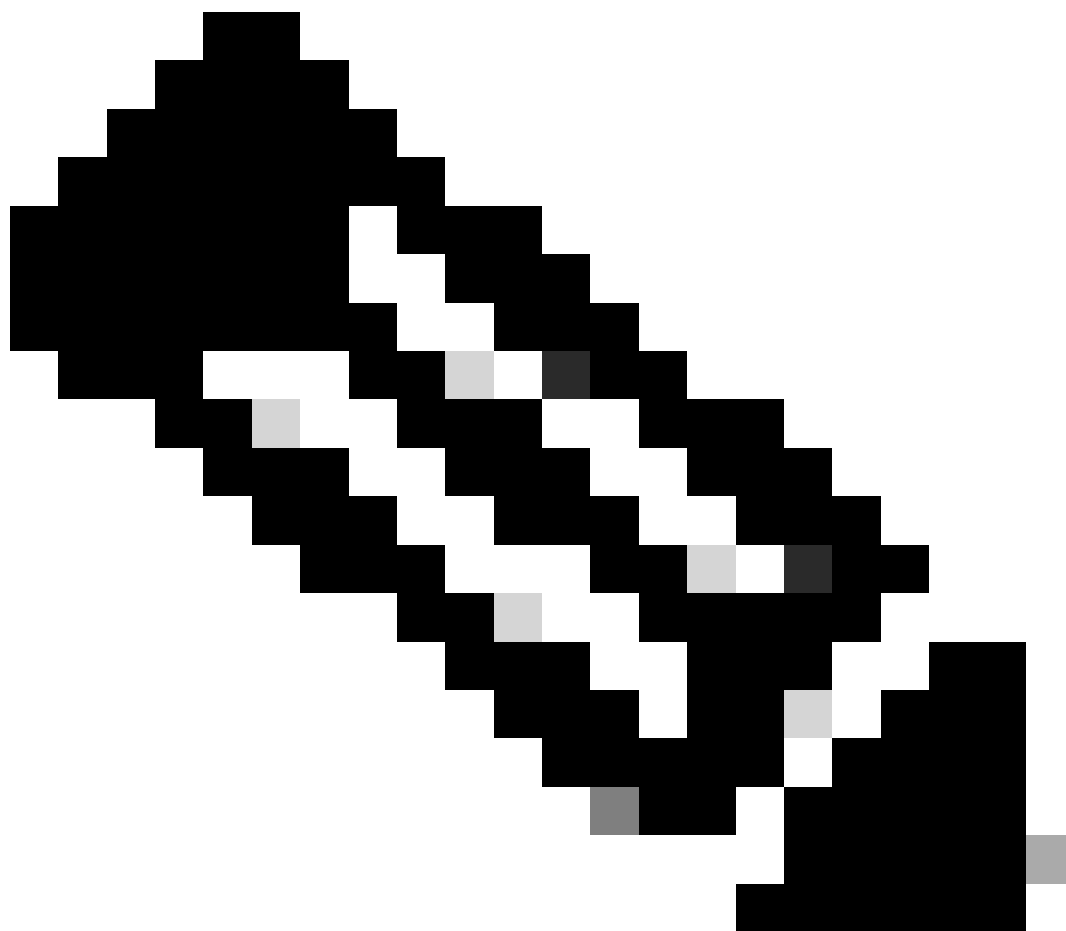
2. Você precisa adicionar uma tag para a solicitação. Na solicitação, você precisa especificar a versão.

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

3. Com o cabeçalho, a solicitação e a versão, o corpo continua com qualquer operação XML que a API tenha:

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

---



Observação: Observe que as solicitações incluem operações diferentes em cada consulta. Não é necessário executar uma solicitação por operação.

---

O próximo exemplo exibe uma solicitação com todas as informações necessárias:

```
<?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>

## Operações com Suporte da API XML

O Cisco IOS XR suporta 5 operações que permitem que os usuários interajam com as informações que o esquema XML possui:

### 1. Operações de Dados Nativos:

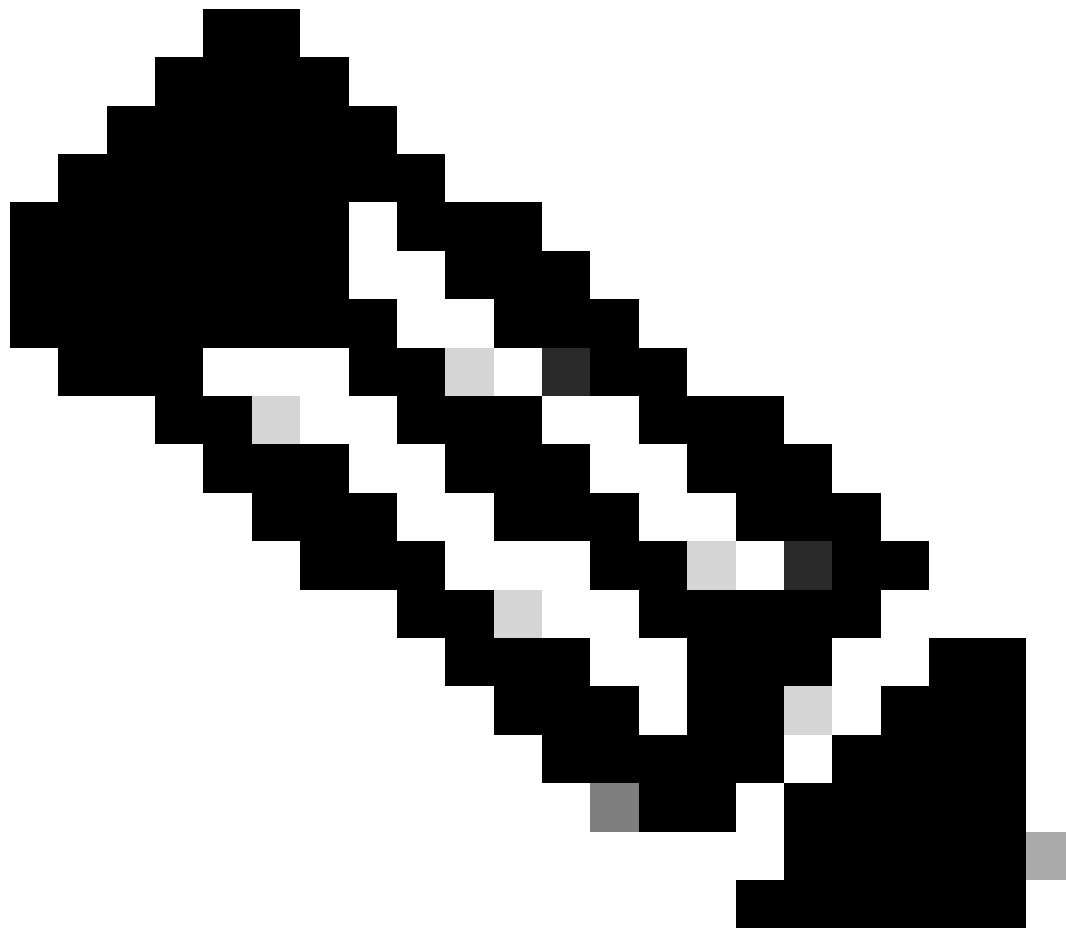
- <Get></Get>: para coletar itens de dados de ação (dados que o leaf tem).
- <Set></Set>: Para configurar, alterar ou adicionar itens de dados.
- <Delete></Delete>: Elimina um ou mais itens de dados.
- <GetVersionInfo></GetVersionInfo>: Operação que recupera os números de versão principal e secundária dos componentes solicitados.
- <GetDataSpaceInfo></GetDataSpaceInfo>: essas marcas operacionais exibem os nomes de folha que são mapeados para os contêineres.

### 2. Operações CLI:

- <CLI></CLI>: marca que nos permite emitir uma solicitação de configuração como um comando CLI.
- <GetNext></GetNext>: Se os dados forem maiores que o tamanho do bloco, esta marca será adicionada à próxima resposta segmentada.

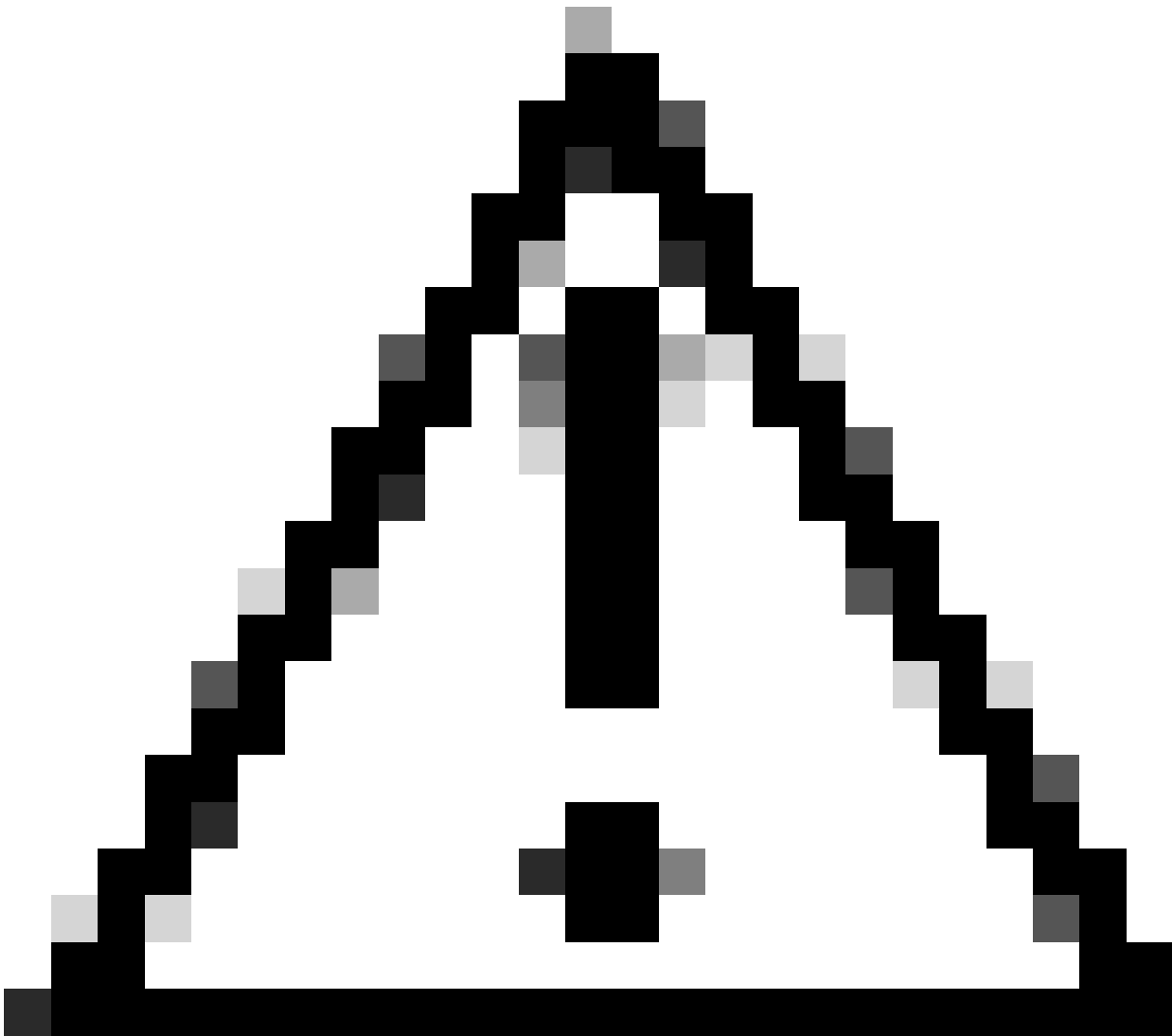
## Serviços oferecidos do esquema XML

A API XML implementa os próximos namespaces:



Observação: caso você precise ver objetos raiz, visite: [Cisco IOS-XR XML Network Management](#).

---



Cuidado: Lembre-se de que cada namespace supervisiona dados e operações diferentes.

- 
- Configuração: operação que permite as ações Get, Set e Delete.
  - Operacional: como os comandos show da CLI.
  - Ação: Suporte à ação Definir. Com essa operação, você pode acessar dados de ação, como comandos clear e outros comandos semelhantes.
  - AdminOperational: acesso aos dados operacionais do administrador.
  - AdminAction: Acesso aos dados da ação administrativa.
  - AdminConfiguration: Acesso aos dados de configuração de administração.

## Criando uma Consulta

Há dois caminhos diferentes para percorrer. Depois de selecionar os namespaces para a consulta, você precisa usar um esquema para trabalhar com:

### 1. Documentação do Esquema XML

Essas opções fornecem uma árvore XML completa que mostra contêineres e folhas. Para obter a documentação, clique em [Esquemas XML CRS](#).

A página de documentação é exibida.

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

Esta página exibe objetos, contêineres e filhos. Cada objeto contém um objeto filho. Se o objeto filho não contiver nenhum outro contêiner, ele será considerado uma folha.

Você pode clicar no filho e as informações para criar uma consulta são exibidas. Por exemplo, para 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>](#)

Na imagem anterior, para qualquer consulta IPv4, a operação de configuração pode ser executada, tendo como alvo o IP do contêiner, no contêiner Cinetd, no contêiner Services e, finalmente, na folha IPv4.



---

Observação: qualquer operação que precise ser uma folha para executar a consulta.

---

A consulta seria a seguinte:

```
<?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. API XML

A API XML está incorporada no Cisco IOS XRI. Para ver a árvore XML, use o comando `show xml schema`. Ao emitir o comando, todo o esquema XML é visto em uma CLI semelhante ao Linux.

Nesta CLI, há as próximas ações:

?: Assim como — help

LS: Liste os contêineres/folhas na ramificação em que o usuário está.

<operation>: ao emitir ? você pode verificar se as operações e os comandos podem ser emitidos:

Algumas operações são:

-config

- adminconfig

-cd

-lista

- oper

-ação

CD: Altere o diretório/contêiner.

O exemplo mostra o seguinte:

```
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>
```

---



Aviso: observe que as credenciais são necessárias para fazer login no dispositivo. Essas credenciais são locais para o dispositivo e exigem perfis root-ls/admin.

---

O próximo exemplo mostra como criar uma consulta usando a API XML. Por exemplo, a consulta precisa verificar se o agente XML está habilitado:

```
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
```

Observe o próximo prompt:

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

O prompt anterior exibe a árvore XML. Com essas informações, a consulta é:

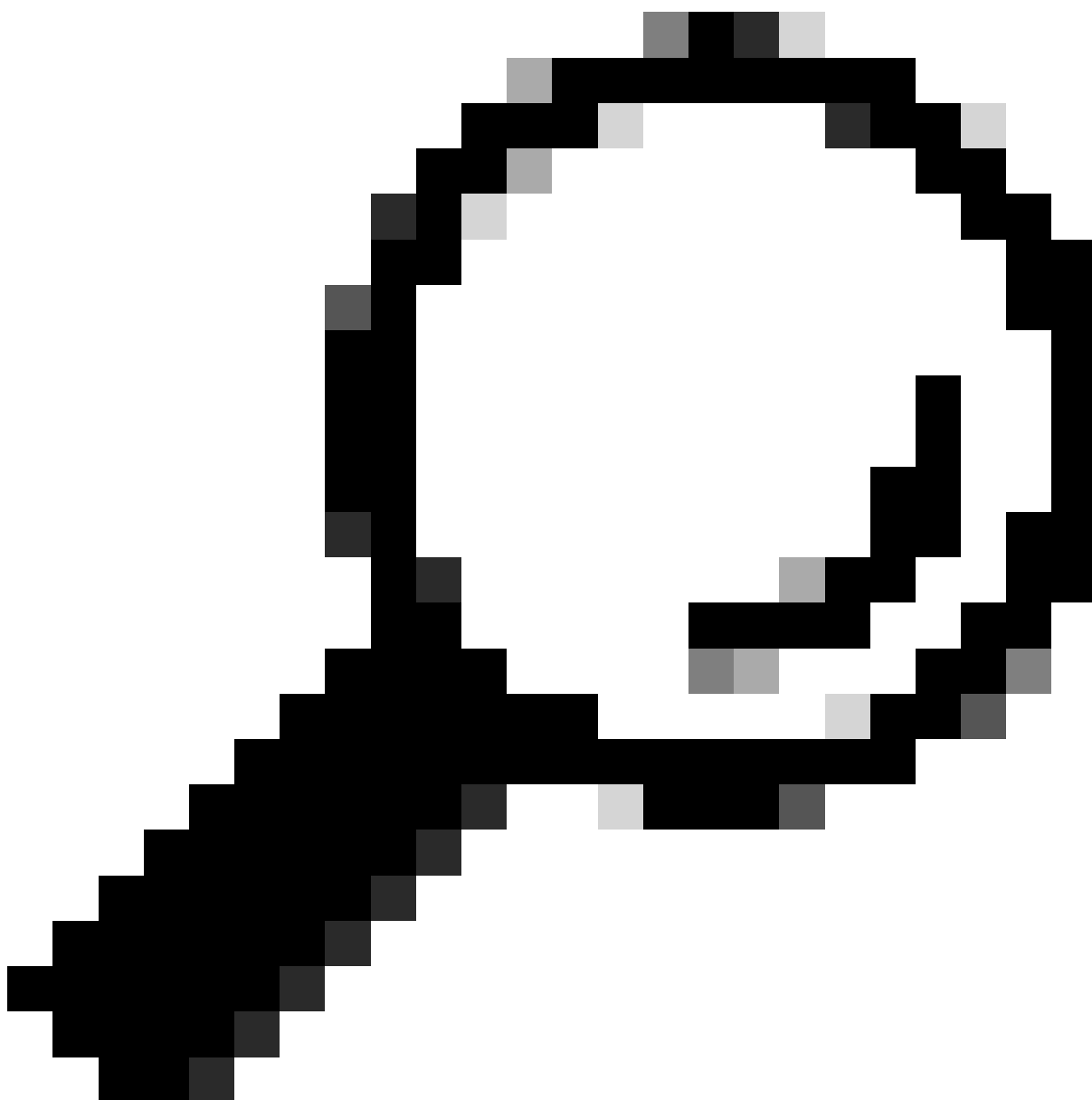
```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
```

```
<Get>
  <Configuration>
    <XML>
      <Agent>
        <TTY></TTY>
      </Agent>
    </XML>
  </Configuration>
</Get>
</Request>
```

## Testando Consultas XML

Depois que a consulta for criada, a próxima etapa será testá-la. Para testá-lo, você pode executar o comando `xml echo format` na CLI. Esta ação pode ser arquivada no mesmo dispositivo.

---



---

Dica: lembre-se de que, após inserir o comando anterior, os usuários podem copiar/colar a consulta e pressionar Enter. O dispositivo exibe a resposta à consulta.

---

```
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
```

O roteador exibe response e ResultSummary.



Observação: essa resposta XML não é exibida nem exige a configuração de memória de aceleração. A resposta, como não está usando nenhum SSH/TTY, é solicitada independentemente do tamanho. Uma coisa é o cliente XML e outra o comando incorporado XML echo, o comportamento pode mudar.

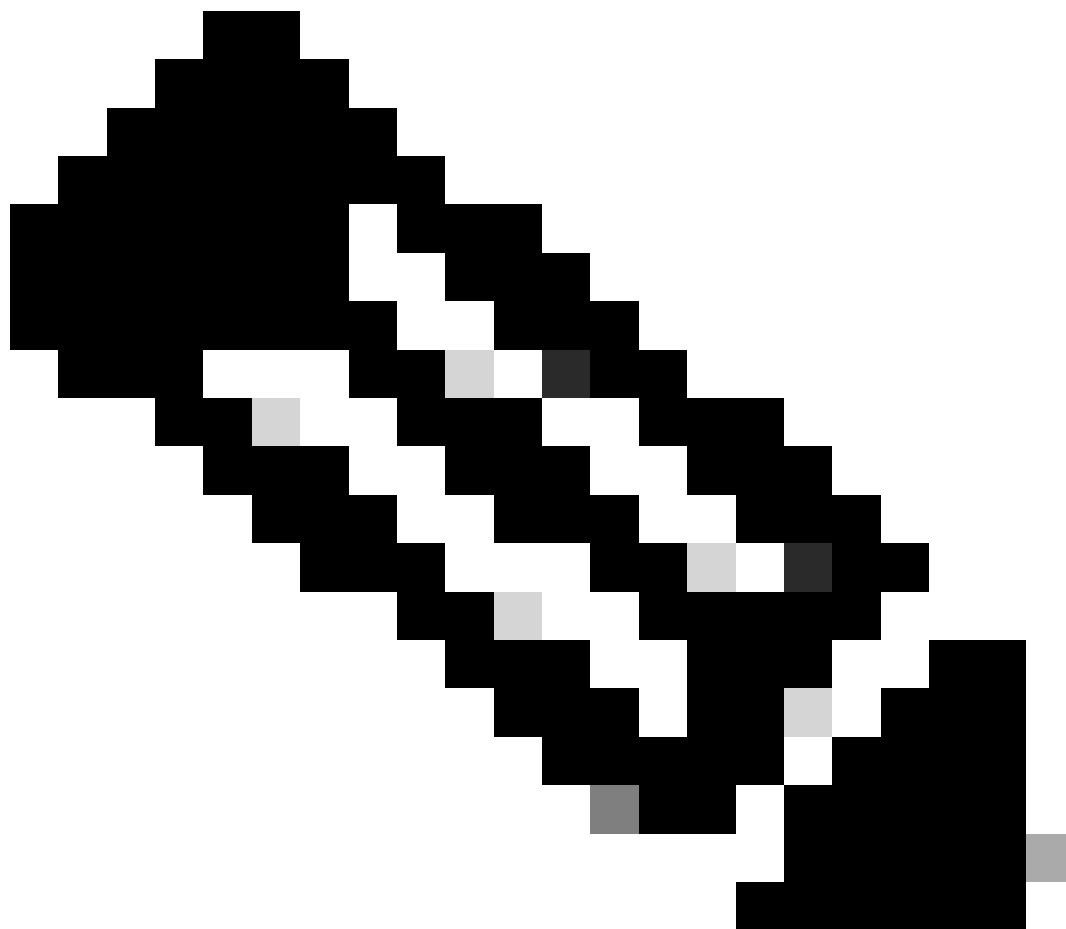
---

## Mergulho profundo em XML no Cisco IOS XR

O Cisco IOS XR é um sistema modular que requer diferentes pacotes para funcionar. O XML é considerado como estando nos subsistemas do plano de gerenciamento.

Quando o XR foi iniciado, havia duas maneiras diferentes de utilizar o XML no dispositivo:

1. Agente CORBA (usando SSL).
2. Agente SSH.



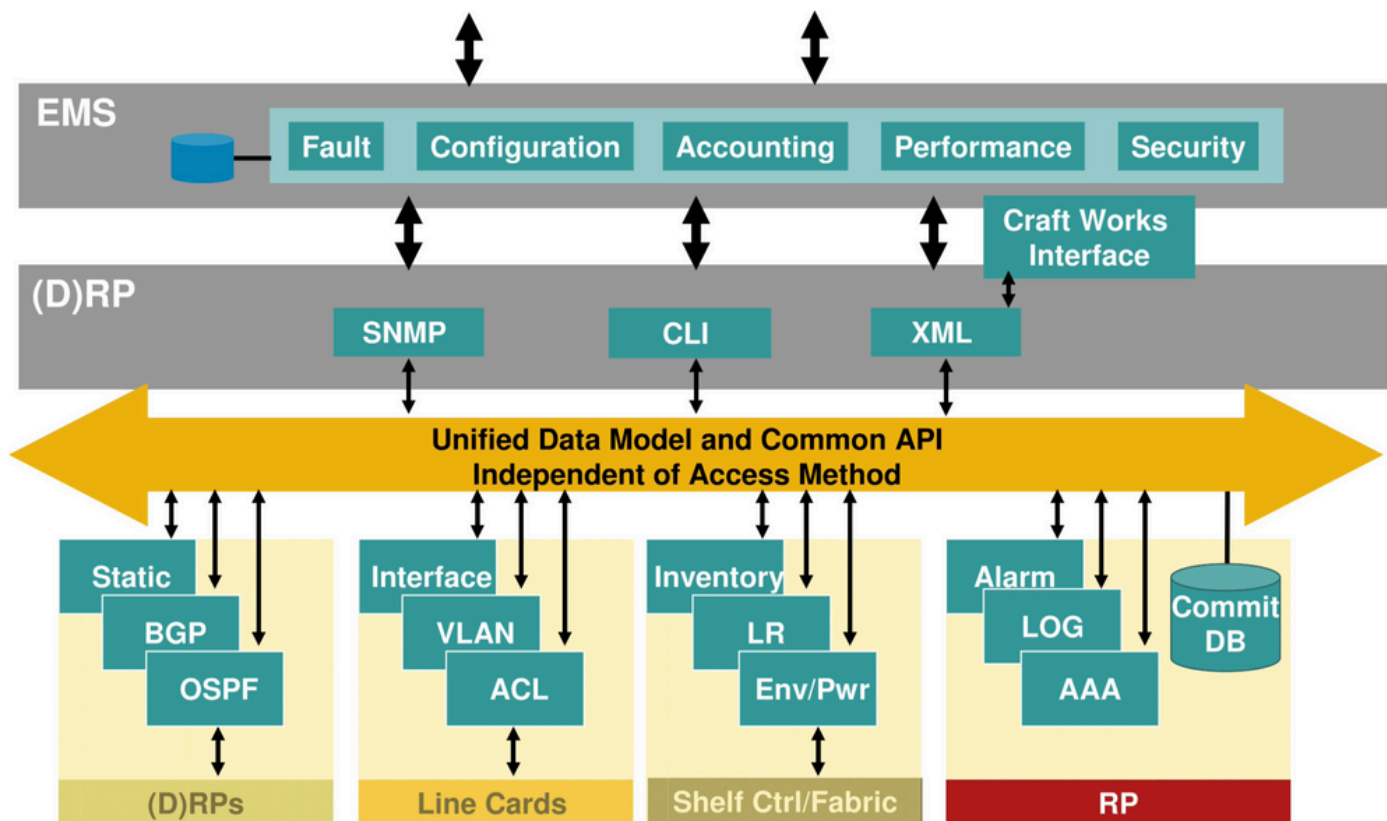
Nota: O CORBA foi substituído após a versão 3.7.

---

O CORBA como um protocolo legado usava SSL, que é encontrado nos Pacotes de segurança. Por outro lado, o método SSH precisa que o Pacote de gerenciabilidade seja usado.

A infra é distribuída desta forma:





A partir da imagem anterior, o processo principal de XML é visto no Router and Switch Processor (RSP). Esses processos têm uma API comum que supervisiona todas as informações dos outros processos no dispositivo, que são:

- Na placa de linha:
  - Coletor de estatísticas
  - Servidor de estatísticas
- A partir do processador do roteador:
  - Stats Manager
  - SysDB.
  - TCP (para API XML é usado 38751 TCP)
  - NetIO



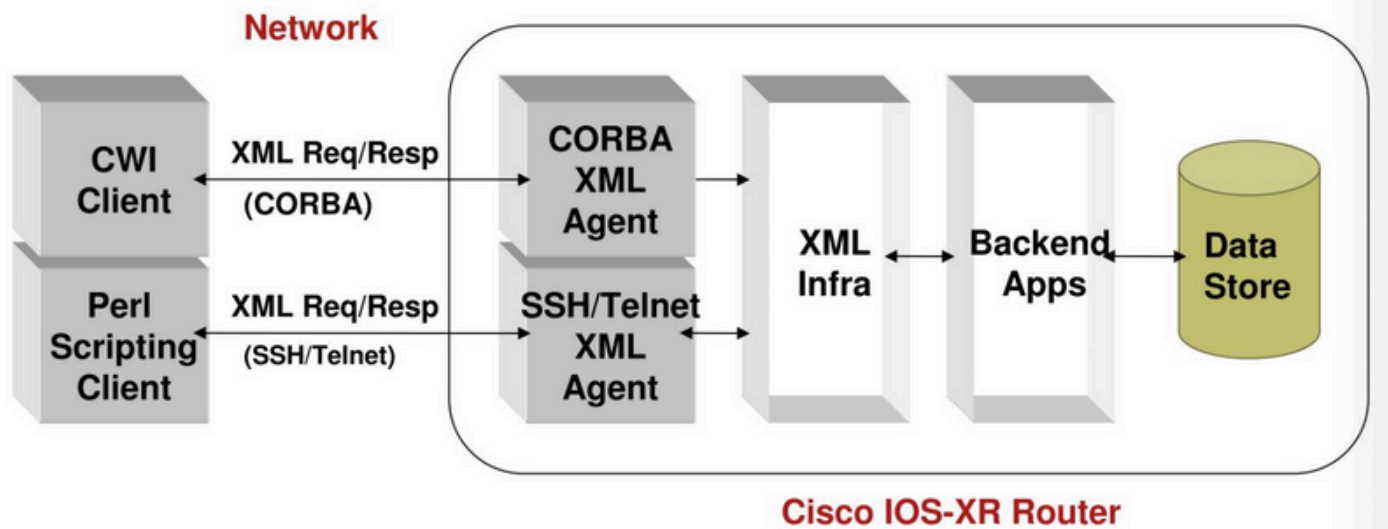
Observação: para ver mais informações sobre isso, verifique os erros XML na seção XR do Cisco IOS.

---

Dependendo da solicitação, um ou mais processos são acionados. Ao desenvolver ferramentas de automação, se a resposta for maior que o tamanho do bloco, a resposta não será bem-sucedida. Se um processo levar muito tempo para responder, ele poderá gerar logs de EDM, desligar ou afetar serviços.

### Interação de API com cliente

A próxima imagem mostra a interação:



Para configurar/habilitar o agente XML no dispositivo, use:

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

O sistema precisa concordar com o cliente nestas áreas:

1. Versões
2. Esquema

As versões podem ser definidas em duas áreas:

- Marca de solicitação:

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

- Marca do componente principal:

Que se aplica ao componente específico:

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



Observação: lembre-se de que você pode verificar a versão do esquema com a operação `GetVersionInfo` aplicada ao contêiner.

- 
- Atualização de versão secundária: Qualquer adição ao esquema XML, como a adição de um novo item de dados.
  - Atualização de versão principal: alterações semânticas, exclusões no esquema ou componente, etc.

Você pode verificar a versão. O próximo exemplo mostra como fazer isso:

---



Observação: as operações XML são GET, SET, etc.

---

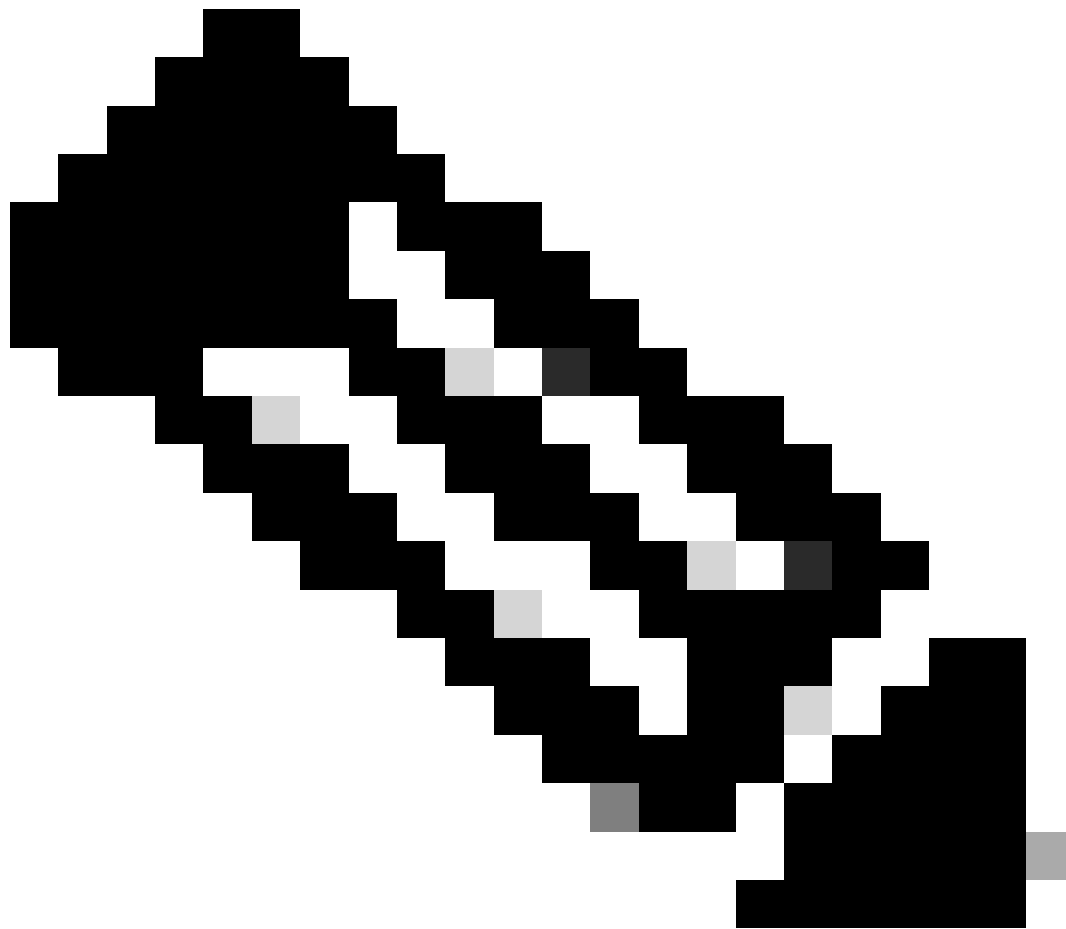
```
<?xml version="1.0" encoding="UTF-8"?>
<Request Major Version="1" MinorVersion="0">
<GetVersionInfo>
<Operation/>
</GetVersionInfo>
</Request>
```

Use o comando `xml echo formal` e adicione a marca `GetVersionInfo` à consulta para localizar a versão que está sendo executada no roteador. como mostrado aqui:

```
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>
```



Observação: a solicitação exibe todas as versões em execução no componente principal e também exibe a versão nos contêineres dentro dele.

---

## Erros XML no Cisco IOS XR

Se o caminho estiver correto, cada solicitação de API XML exibirá as informações solicitadas.

### Quando a consulta está incorreta

O roteador exibe três mensagens diferentes:

- ItemNãoLocalizado

Essa mensagem é exibida toda vez que uma operação GET tem uma resposta vazia.

- ItemNãoEncontradoAbaixo

A operação GET não contém esta operação no Esquema XML.

- NãoEncontrado

O elemento solicitado não pode ser encontrado no nível do elemento.

## Tipos de erro

1. Transport: os erros nesta categoria incluem qualquer coisa entre a comunicação do agente/cliente XML. Isso significa que qualquer interação ou problema de SSH pode acontecer no transporte. Portanto, para verificar esses tipos de problemas, sugere-se verificar os rastreamentos SSH para verificar qualquer problema com autenticação, porta, etc.

2. Analisador XML: Qualquer problema com o formato e a sintaxe, problemas na resposta enviada ou na consulta. Esses problemas normalmente enviam o motivo da falha quando ocorre um erro.

Por exemplo:

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

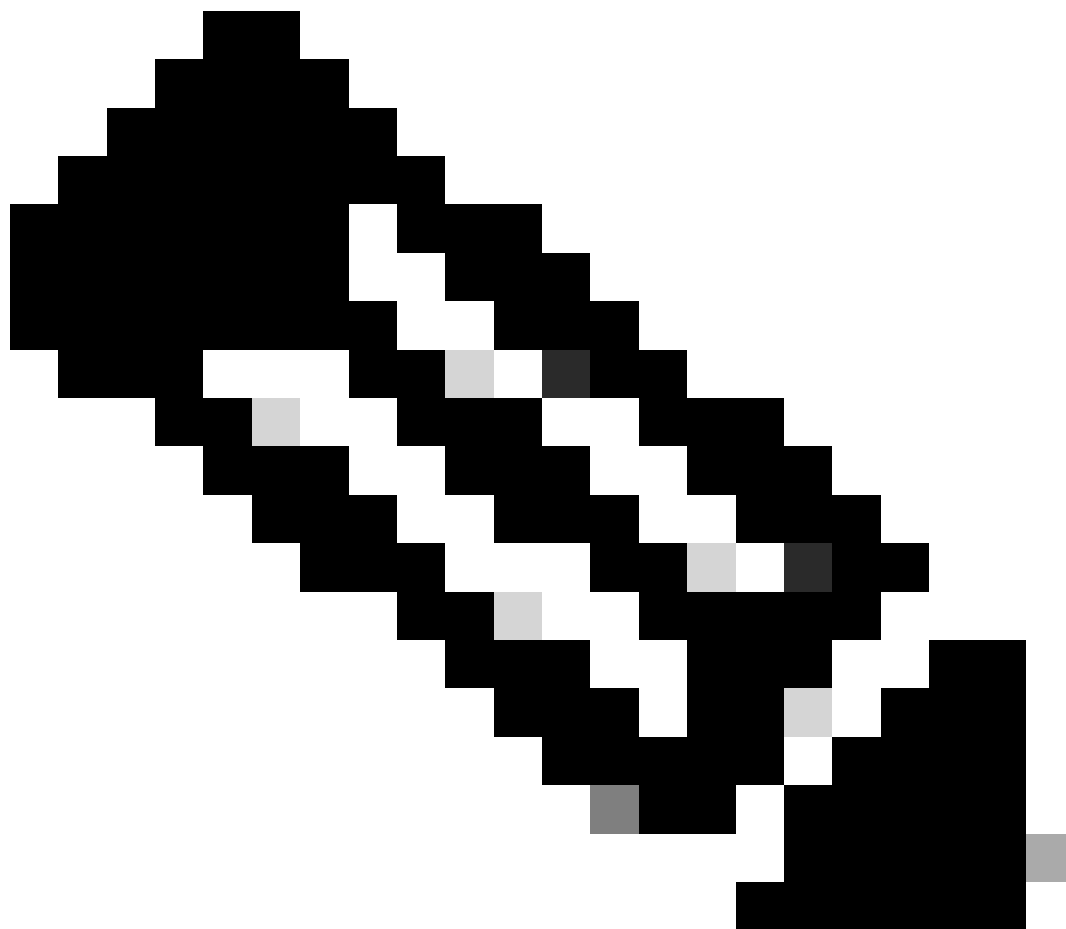
3. Esquema XML: Qualquer esquema diferente do esquema do roteador. Para resolver esses problemas, verifique o esquema e as versões da CLI.

Por exemplo:

```
ErrorCode="0x43688400" ErrorMessage="&apos;XMLMDA&apos; detected the &apos;warning&apos; condition &apos;An
```

4. Processamento da Operação: Quando você configura o dispositivo, esses problemas podem ocorrer. Para resolver esses problemas, você precisa solucionar problemas dos processos, por exemplo, commit, sysdb, etc.





Observação: as informações de erro são adicionadas no nível do elemento de operação. Ele é codificado na forma de atributos `ErrorCode` (int de 32 bits) e `Errormsg`.

---

## Outras recomendações

Outras técnicas úteis:

1. Curinga: Este é o também conhecido como Consultas Específicas.
2. Criação de Lotes: Combinação de várias técnicas ou operações em uma única solicitação (operações de melhor esforço).
3. Filtragem personalizada: Se o esquema permitir, para ajudar na seleção de linhas nas tabelas.

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.