

Dépannage de XML pour Cisco IOS XR

Table des matières

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Problèmes de mémoire de limitation](#)

[Description supplémentaire du problème](#)

[Solution](#)

[XML dans Cisco IOS XR](#)

[Utilisation de l'API XML](#)

[Opérations prises en charge par l'API XML](#)

[Services de schéma XML offerts](#)

[Création d'une requête](#)

[Test des requêtes XML](#)

[Approfondissement XML dans Cisco IOS XR](#)

[Interaction API avec le client](#)

[Erreurs XML dans Cisco IOS XR](#)

[Lorsque la requête est incorrecte](#)

[Types d'erreurs](#)

[Autres recommandations](#)

Introduction

Ce document décrit les problèmes d'étranglement de mémoire XML rencontrés dans les demandes de service et les outils Cisco.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Plate-forme Cisco IOS® XR
- Infrastructure XML (Extensible Markup Language)
- Architecture CORBA (Common Object Request Broker Architecture)
- Système de gestion des éléments (EMS)
- Gestionnaire de données externes (EDM)

Composants utilisés

Les informations contenues dans ce document sont basées sur ASR9000.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Problèmes de mémoire de limitation

Un dossier a été ouvert avec ces questions :

1. Quelle est la mémoire maximale configurable sur le routeur ?
2. Existe-t-il un moyen de supprimer les réponses XML volumineuses ?

Réponse :

1. Modification de la mémoire maximale en fonction de la version exécutée par le RSP/noeud (versions pour cXR et eXR)

En version 64 bits (eXR). Vous avez une taille de mémoire d'accélération :

```
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)
```

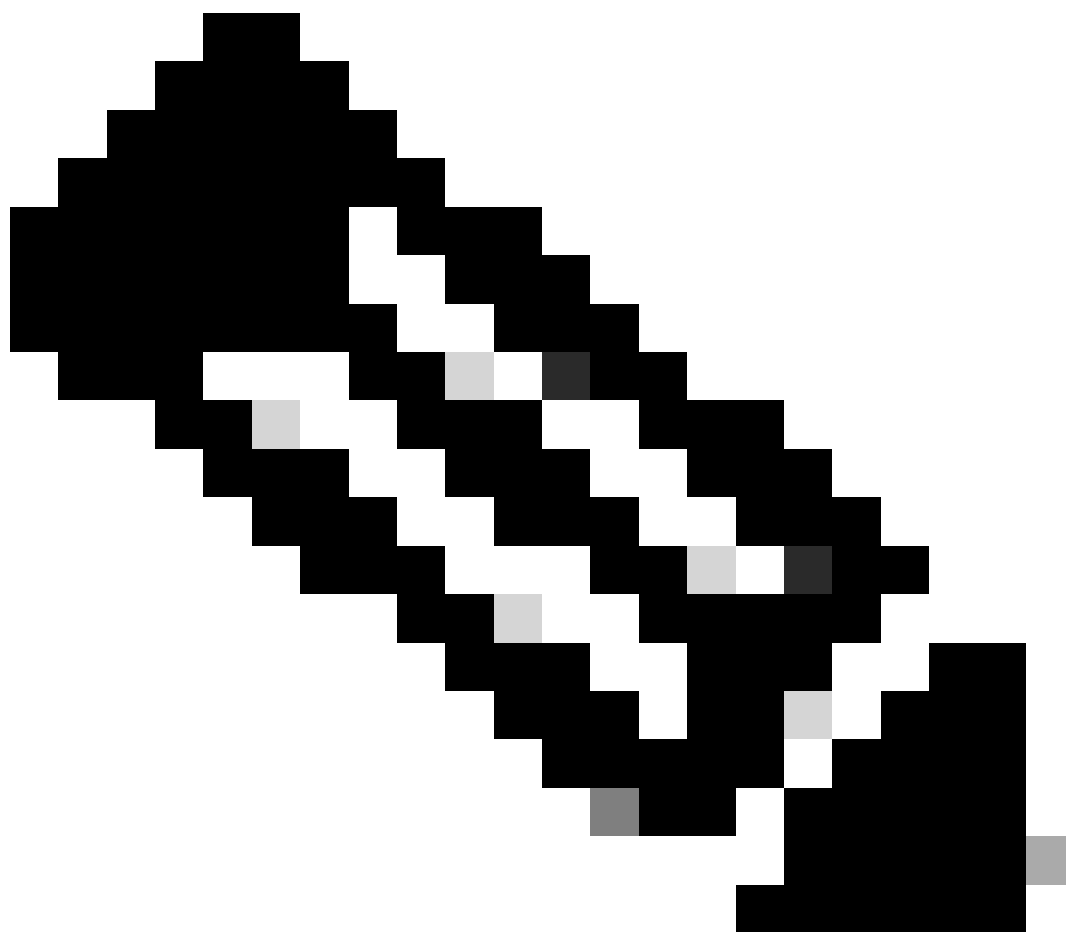
En version 32 bits (cXR). Vous avez une autre limite :

```
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)



Remarque : la valeur par défaut dans les deux versions est 300 Mo

2. Il existe différentes manières de rompre la réponse XML :

- Itérateurs :

Lorsque des itérateurs sont configurés, vous segmentez la réponse totale XML en taille de fenêtre spécifique. La fenêtre a la taille de l'itérateur. Par exemple, si la réponse est 1 Go et

que la taille de l'itérateur est de 500 Mo, vous segmentez le XML en 2 réponses.

Cette méthode modifie la réponse en ajoutant (en fonction de la taille de l'itérateur) des opérations GetNext (ce dont la réponse a besoin).

Il peut y avoir jusqu'à 10 itérateurs par session.

- Limitation (vitesse de traitement) :

Cette fonctionnalité limite la mémoire utilisée par le processus XML. Si un processus dépasse la mémoire, il répond avec le message d'erreur : "L'étranglement sur l'utilisation de la mémoire a été atteint".

- Diffusion :

Tout comme les itérateurs, la diffusion en continu segmente la réponse XML en fenêtres spécifiques. La différence est la réponse, elle supprime l'opération GetNext et l'ID d'itérateur. Le code XML envoie les packages diffusés en continu et le client génère la réponse lorsque le flux se termine.

Description supplémentaire du problème

Pour les cas d'utilisation où l'automatisation est nécessaire, utilisez l'outil [pyIOSXR](#). Cet outil d'automatisation est un agent XML, il permet d'émettre quelques commandes show et, de manière générale, se connecte au périphérique.

Chaque fois que vous envoyez une requête importante avec cet agent, une erreur s'affiche :



Remarque : seuls les utilisateurs Cisco enregistrés peuvent accéder aux informations et aux outils Cisco internes.

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

Avec un message d'erreur affichant "optimiser la requête pour interroger des données plus petites", vous pouvez configurer l'API XML de l'agent avec des itérateurs. Cela permet, en théorie, de segmenter la réponse.

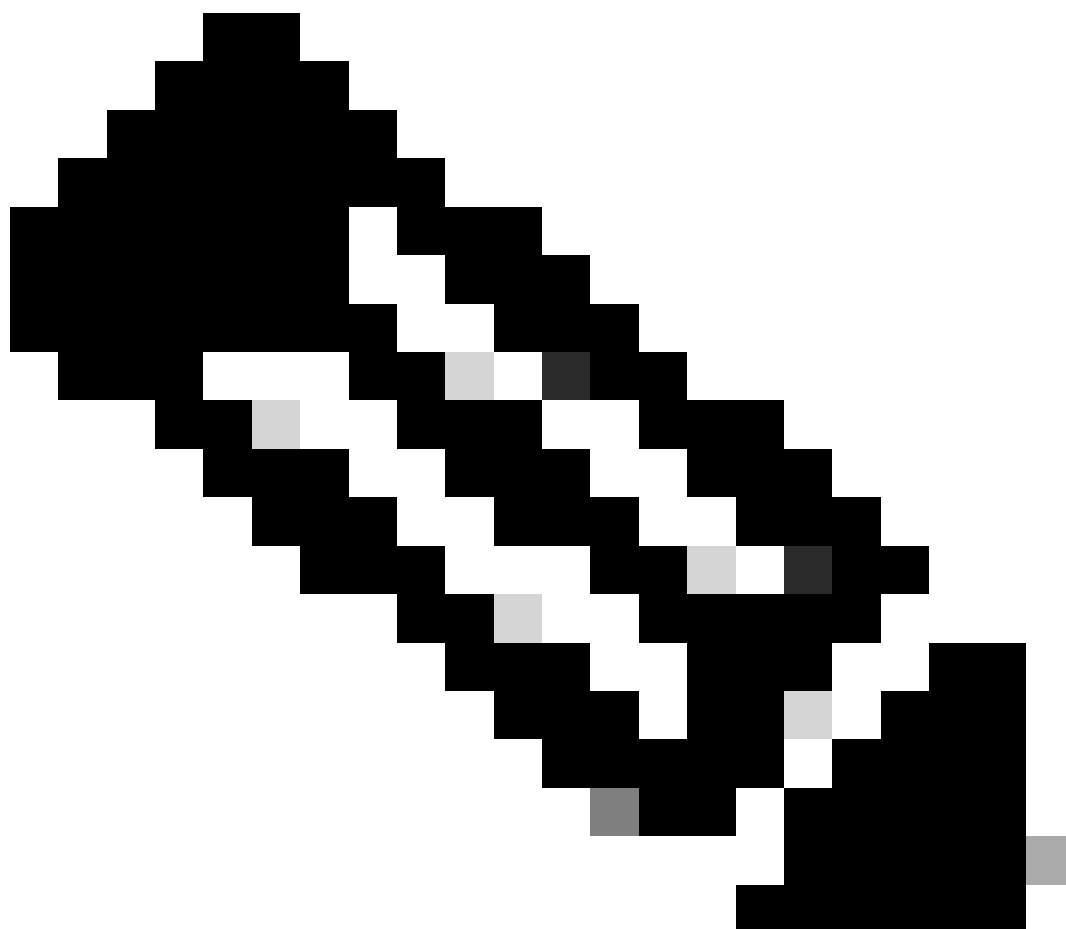
Lorsque les itérateurs ont été configurés, le même message d'erreur s'est affiché : « ERREUR DE RÉPONSE...Veuillez optimiser la requête pour demander des données plus petites ».

Lorsqu'une erreur s'affiche, l'étape suivante consiste à comprendre pourquoi la fonctionnalité, en l'occurrence Iterators, n'a pas fonctionné pour la requête.

Le [pyIOSXR](#) recommande que l'agent XML soit correctement activé dans le périphérique, ce qui

signifie que l'API ne permet pas l'utilisation d'itérateurs.

L'étape suivante consiste à tester la deuxième option : la diffusion en continu.



Note : pyIOSXR ne nous permet pas de travailler avec des en-têtes différents autres que la requête XML de base qui ont les éléments suivants :

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

Les flux et les itérateurs ajoutent un en-tête aux requêtes. Le streaming ajoute un ID de flux qui

aide le système à générer la réponse complète. Les itérateurs ajoutent GET_Next et Iterator_ID.

Les tests de diffusion en continu ont également montré le même message d'erreur que les itérateurs.

Solution

Dans la section Plus de recommandations pour effectuer des requêtes sur le périphérique, il y a plus d'outils pour résoudre ces problèmes. Le caractère générique est l'un d'eux. Le caractère générique est la solution pour la limite de limitation de la mémoire.

Le caractère générique crée une requête spécifique pour éviter de demander des informations inutiles. Par exemple, pour les informations BGP, utilisez la commande show route bgp au lieu de la commande générique show route. Cet exemple s'applique aux requêtes et à la logique XML. La demande d'informations en masse au système peut générer des problèmes de mémoire et de traitement.

Lorsque le dossier a été ouvert, cette requête a été utilisée :

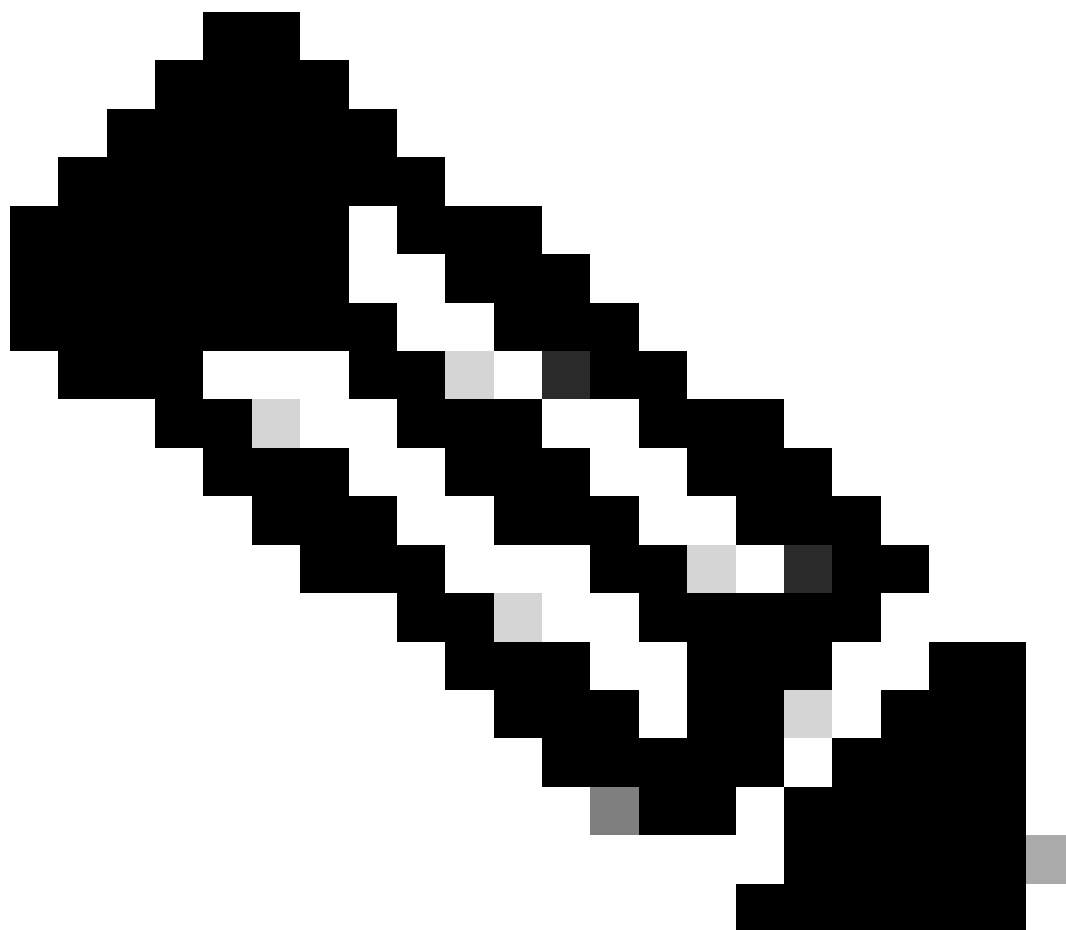
```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get>
    <Operational>
      <BGP MajorVersion="36" MinorVersion="1">
        <InstanceTable>
          <Instance>
            <Naming>
              <InstanceName>
                DEFAULT
              </InstanceName>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

Cette requête cible les tables BGP complètes. Pour cette requête non spécifique, la réponse était de près de 2,2 Go, par conséquent, la limite de limitation de la mémoire est atteinte.

Pour le résoudre, une requête spécifique est requise, ce qui permet au système de traiter la requête et de renvoyer les informations.

XML dans Cisco IOS XR



Remarque : si vous avez besoin de plus d'informations, de spécifications ou d'autres commandes, cliquez sur le lien suivant Guide de programmation XML : [Guide de programmation XML XR](#)

Le code XML définit la manière dont les données sont affichées et structurées. Il s'agit d'un moyen d'analyser ce que l'ordinateur comprend comme des bits et d'afficher des informations structurées et normalisées.

XML a cette structure :

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

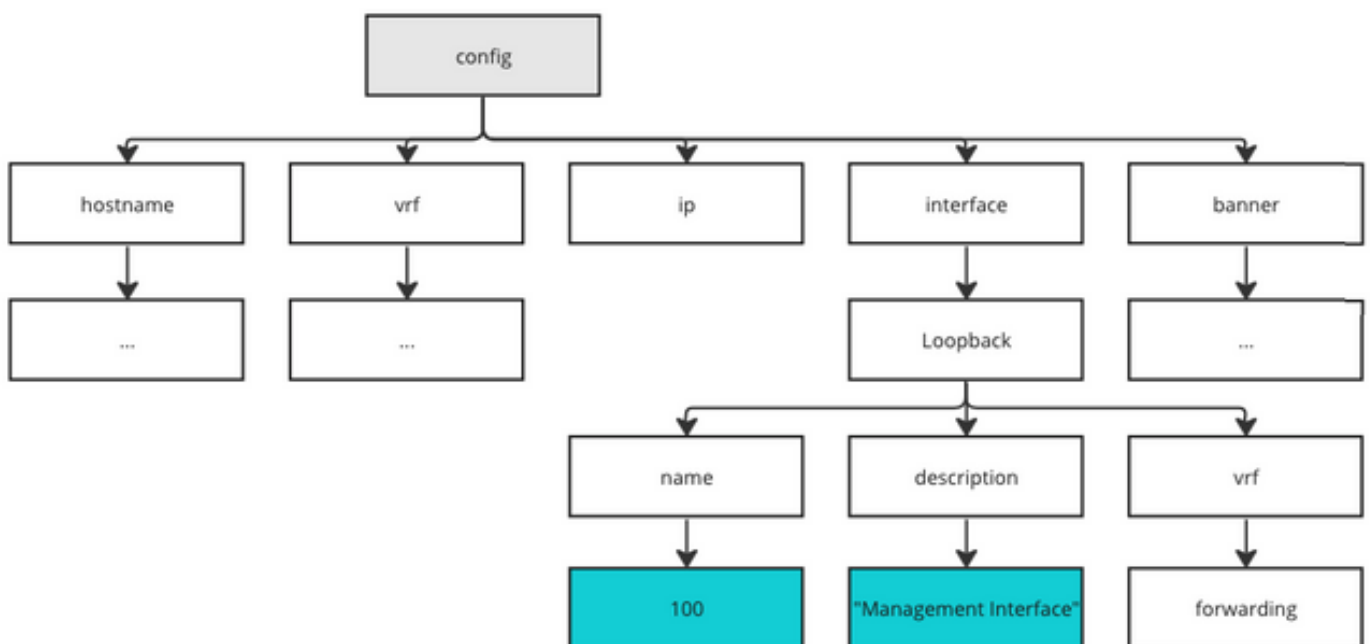
Toutes les balises comportent deux parties, la balise d'ouverture (<init>) et la balise de fermeture (</init>). Si cette structure n'est pas utilisée, XML ne peut pas comprendre où se termine une balise.

XML est constitué de deux entités :

- Conteneurs
- Feuilles

Une analogie pour ces données structurées est la comparaison des données XML sous forme d'arborescence. Un conteneur est une branche et chaque branche a des feuilles coupées. Les feuillets ne contiennent que des informations.

Par exemple, la racine de l'arbre suivant est grise, les contenants blancs et les feuilles bleues.

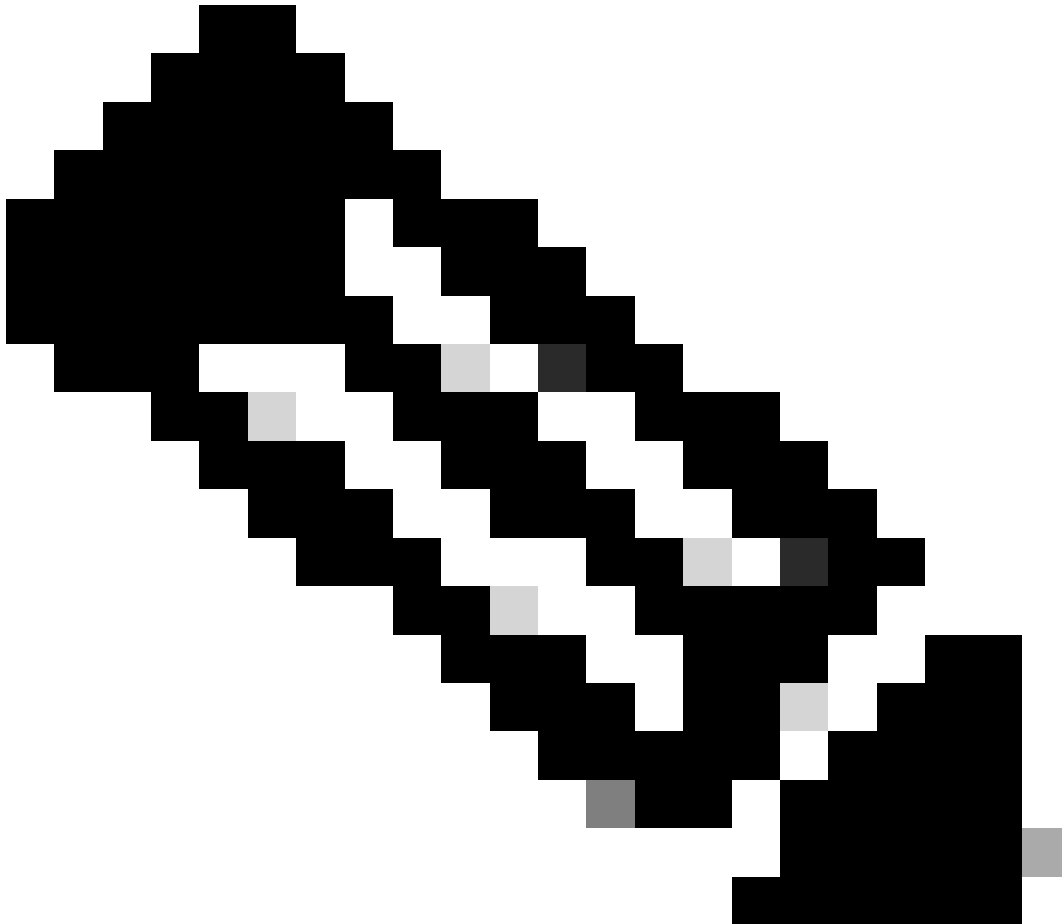


Utilisation de l'API XML

Pour tester/utiliser l'API XML, la première chose dont vous avez besoin est une requête.

1. La requête a un en-tête :

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



Remarque : il s'agit de la seule balise qui n'a pas besoin d'une balise de fermeture.

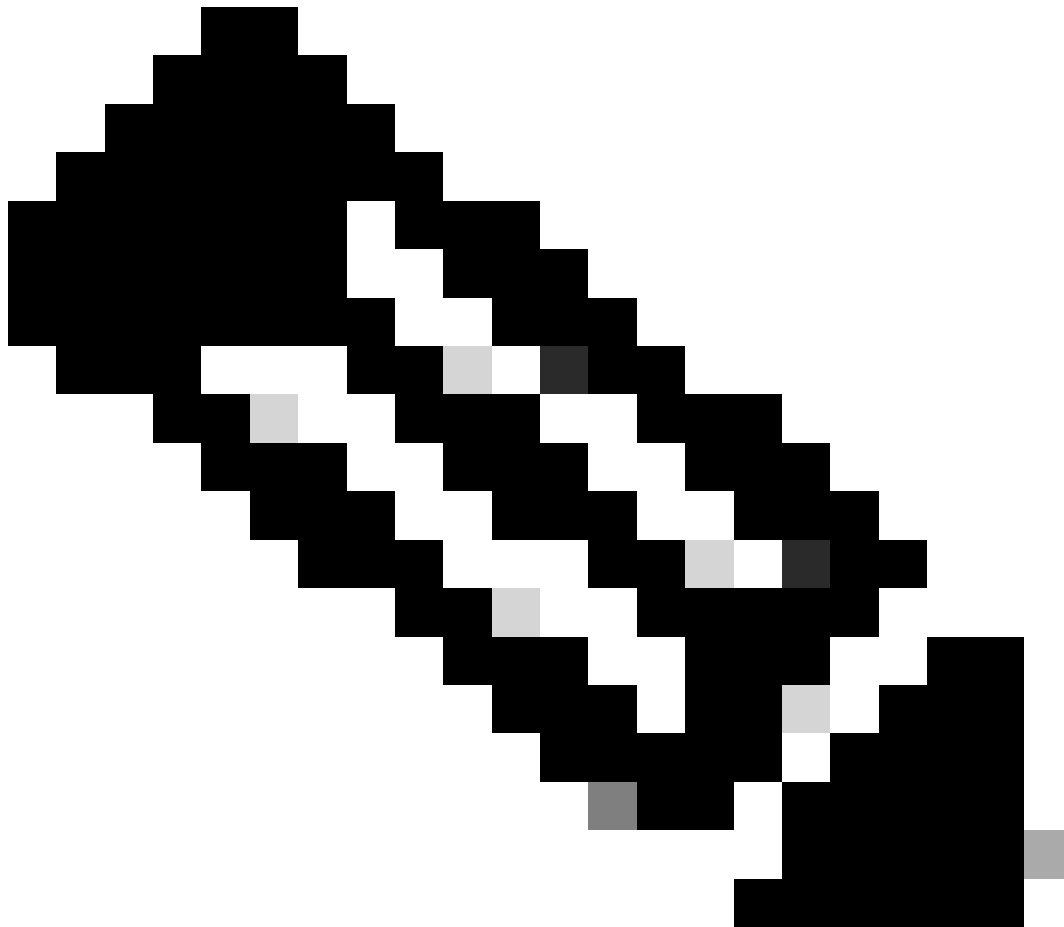
2. Vous devez ajouter une balise pour la demande. Dans la demande, vous devez spécifier la version.

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

3. Avec l'en-tête, la demande et la version, le corps continue avec toute opération XML que l'API a

:

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



Remarque : notez que les requêtes incluent des opérations différentes dans chaque requête. Il n'est pas nécessaire d'effectuer une requête par opération.

L'exemple suivant affiche une demande avec toutes les informations requises :

```
<?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>
```

Opérations prises en charge par l'API XML

Cisco IOS XR prend en charge 5 opérations qui permettent aux utilisateurs d'interagir avec les informations contenues dans le schéma XML :

1. Opérations de données natives :

- `<Get></Get>` : pour collecter les éléments de données d'action (données dont dispose le leaf).
- `<Set></Set>` : pour configurer, modifier ou ajouter des éléments de données.
- `<Delete></Delete>` : supprime un ou plusieurs éléments de données.
- `<GetVersionInfo></GetVersionInfo>` : opération qui récupère les numéros de version principal et secondaire des composants demandés.
- `<GetDataSpaceInfo></GetDataSpaceInfo>` : ces balises opérationnelles affichent les noms de feuille qui sont mappés aux conteneurs.

2. Opérations CLI :

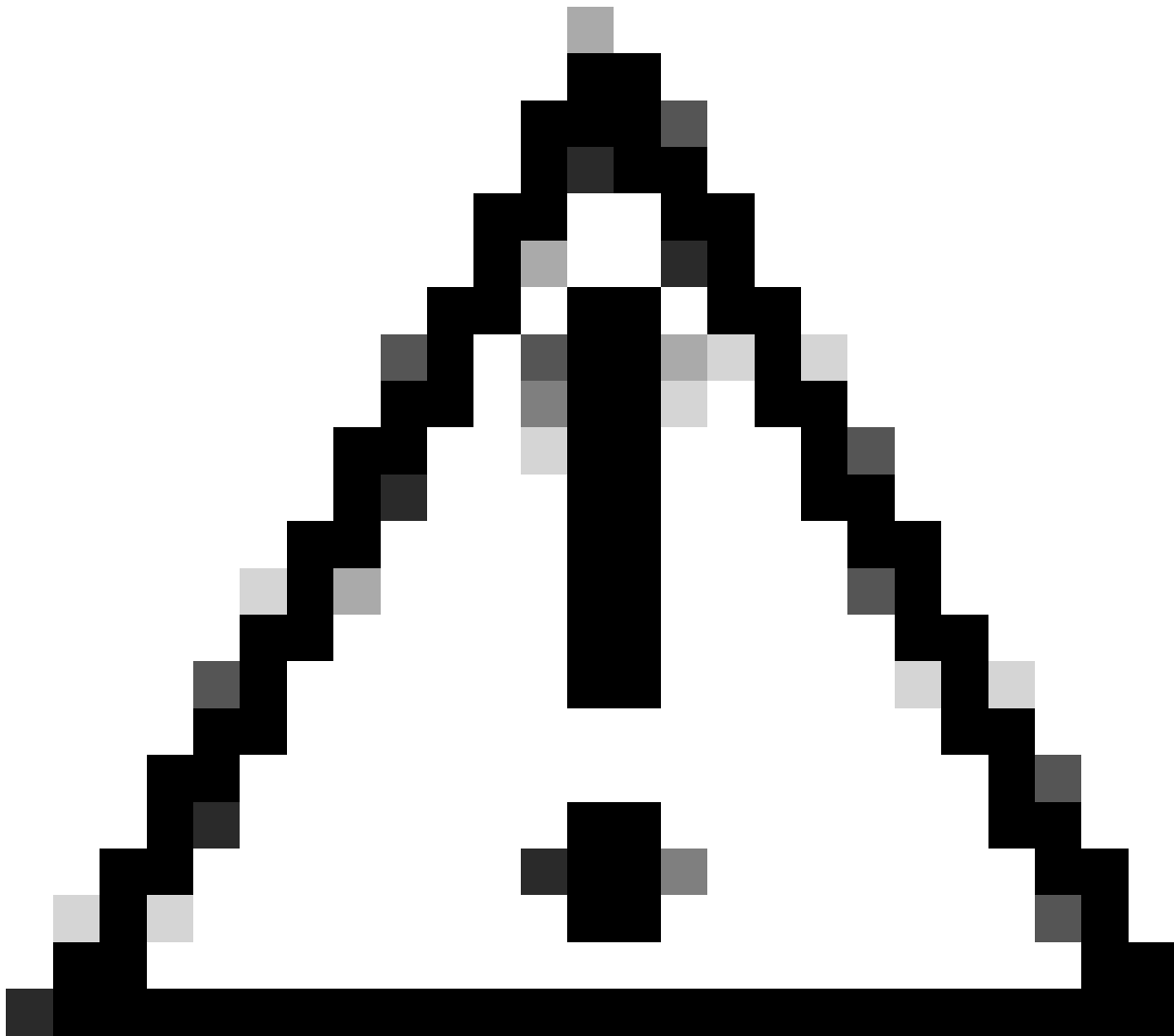
- `<CLI></CLI/>` : balise permettant d'émettre une demande de configuration sous la forme d'une commande CLI.
- `<GetNext></GetNext/>` : si les données sont plus volumineuses que la taille de bloc, cette balise est ajoutée à la réponse segmentée suivante.

Services de schéma XML offerts

L'API XML implémente les espaces de noms suivants :



Remarque : si vous avez besoin de voir les objets racine, visitez : [Cisco IOS-XR XML Network Management](#).



Attention : chaque espace de noms supervise différentes données et opérations.

-
- Configuration : opération qui permet d'exécuter les actions Get, Set et Delete.
 - Opérationnel : comme les commandes show CLI.
 - Action : prenez en charge l'action Définir. Cette opération vous permet d'accéder à des données d'action, telles que des commandes clear et d'autres commandes similaires.
 - AdminOperational : accès aux données opérationnelles Admin.
 - AdminAction : accès aux données d'action d'administration.
 - AdminConfiguration : accès aux données de configuration d'administration.

Création d'une requête

Il y a deux chemins différents pour voyager. Une fois que vous avez sélectionné les espaces de noms pour la requête, vous devez utiliser un schéma pour travailler avec :

1. Documentation du schéma XML

Ces options fournissent une arborescence XML complète qui affiche les conteneurs et les leafs. Pour obtenir de la documentation, cliquez sur [Schémas XML CRS](#).

La page de documentation s'affiche.

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

Cette page affiche les objets, les conteneurs et les enfants. Chaque objet contient un objet enfant. Si l'objet enfant ne contient aucun autre conteneur, il est considéré comme un noeud leaf.

Vous pouvez cliquer sur l'enfant et les informations pour créer une requête s'affichent. Par exemple pour 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>](#)

Dans l'image précédente, pour toute requête IPv4, l'opération de configuration peut être exécutée, en ciblant l'adresse IP du conteneur, dans le conteneur Cinetd, dans le conteneur Services, puis, enfin, dans le leaf IPv4.



Remarque : toute opération qui doit être un leaf pour exécuter la requête.

La requête serait la suivante :

```
<?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

L'API XML est intégrée à Cisco IOS XRI. Pour afficher l'arborescence XML, utilisez la commande `show xml schema`. En émettant la commande, le schéma XML entier est vu dans une CLI similaire à Linux.

Dans cette interface de ligne de commande, vous pouvez effectuer les actions suivantes :

? : Exactement comme `-help`

LS : répertoriez les conteneurs/leafs de l'utilisateur de la succursale.

<operation> : en exécutant ? vous pouvez vérifier que les opérations et les commandes peuvent être exécutées :

Certaines opérations sont les suivantes :

-configuration

- adminconfig

-cd

- liste

- oper

-action

CD : Modifier le répertoire/conteneur.

L'exemple ci-dessous :

```
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>
```



Avertissement : notez que des informations d'identification sont nécessaires pour vous connecter au périphérique. Ces informations d'identification sont locales au périphérique et nécessitent des profils root-Is/admin.

L'exemple suivant montre comment créer une requête à l'aide de l'API XML. Par exemple, la requête doit vérifier si l'agent XML est activé :

```
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
```

Notez l'invite suivante :

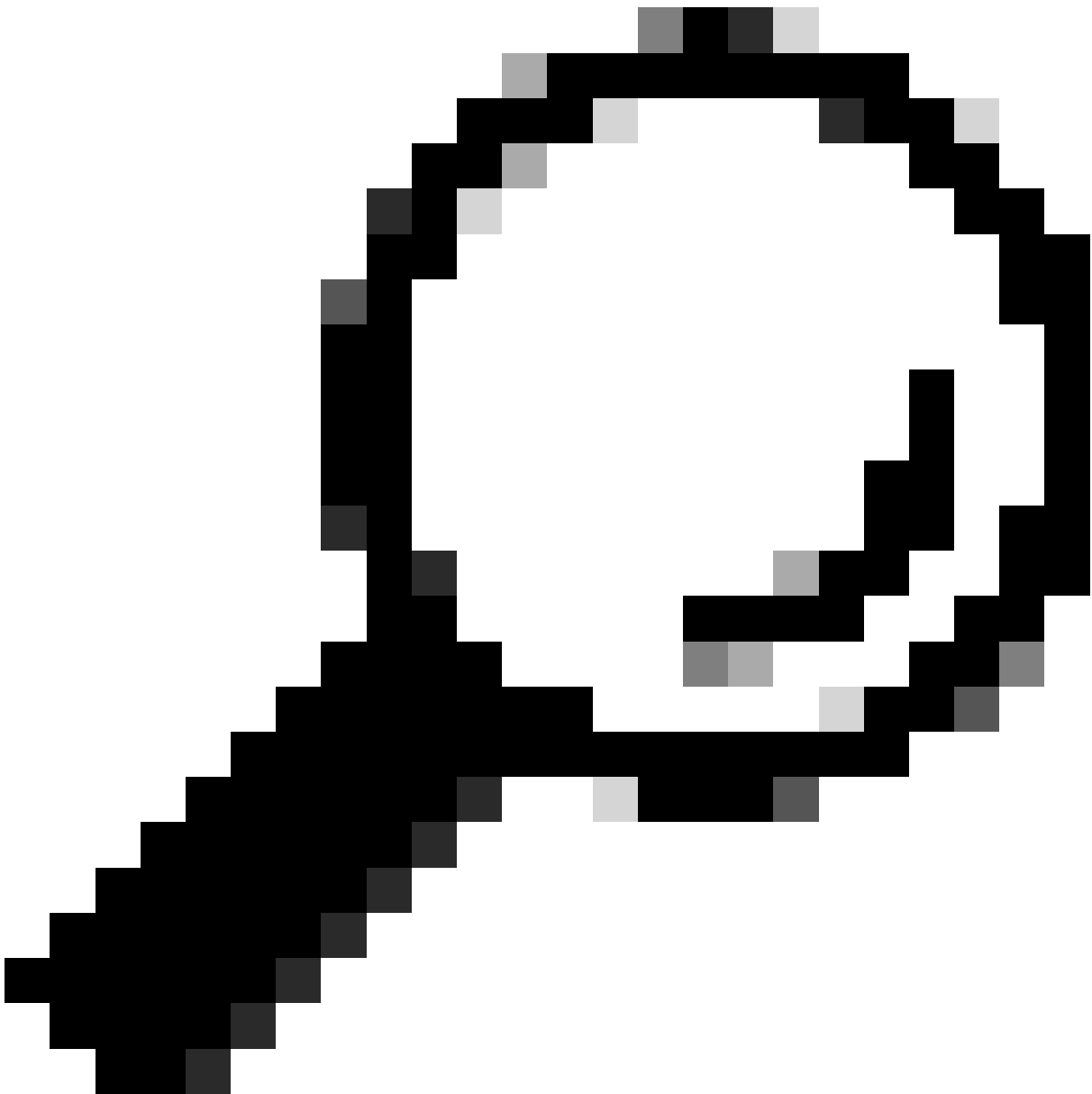
```
xml-schema[config]:XML.Agent.TTY>
```

L'invite précédente affiche l'arborescence XML. Avec ces informations, la requête est :

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

Test des requêtes XML

Une fois la requête créée, l'étape suivante consiste à la tester. Pour le tester, vous pouvez émettre la commande `xml echo format` dans l'interface de ligne de commande. Cette action peut être archivée sur le même périphérique.



Conseil : notez qu'après avoir entré la commande précédente, les utilisateurs peuvent copier/coller la requête et appuyer sur Entrée. Le périphérique affiche la réponse à la requête.

```
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
```

Le routeur affiche la réponse et le résumé des résultats.



Remarque : cette réponse XML n'affiche pas et ne nécessite aucune mémoire d'accélération pour être configurée. La réponse, car elle n'utilise pas de SSH/TTY, est demandée quelle que soit la taille. D'une part, le client XML et, d'autre part, la commande d'écho XML intégrée, le comportement peut changer.

Approfondissement XML dans Cisco IOS XR

Cisco IOS XR est un système modulaire qui nécessite différents packages pour fonctionner. XML est considéré comme faisant partie des sous-systèmes du plan de gestion.

Lorsque XR a démarré, il y avait 2 façons différentes d'utiliser XML dans le périphérique :

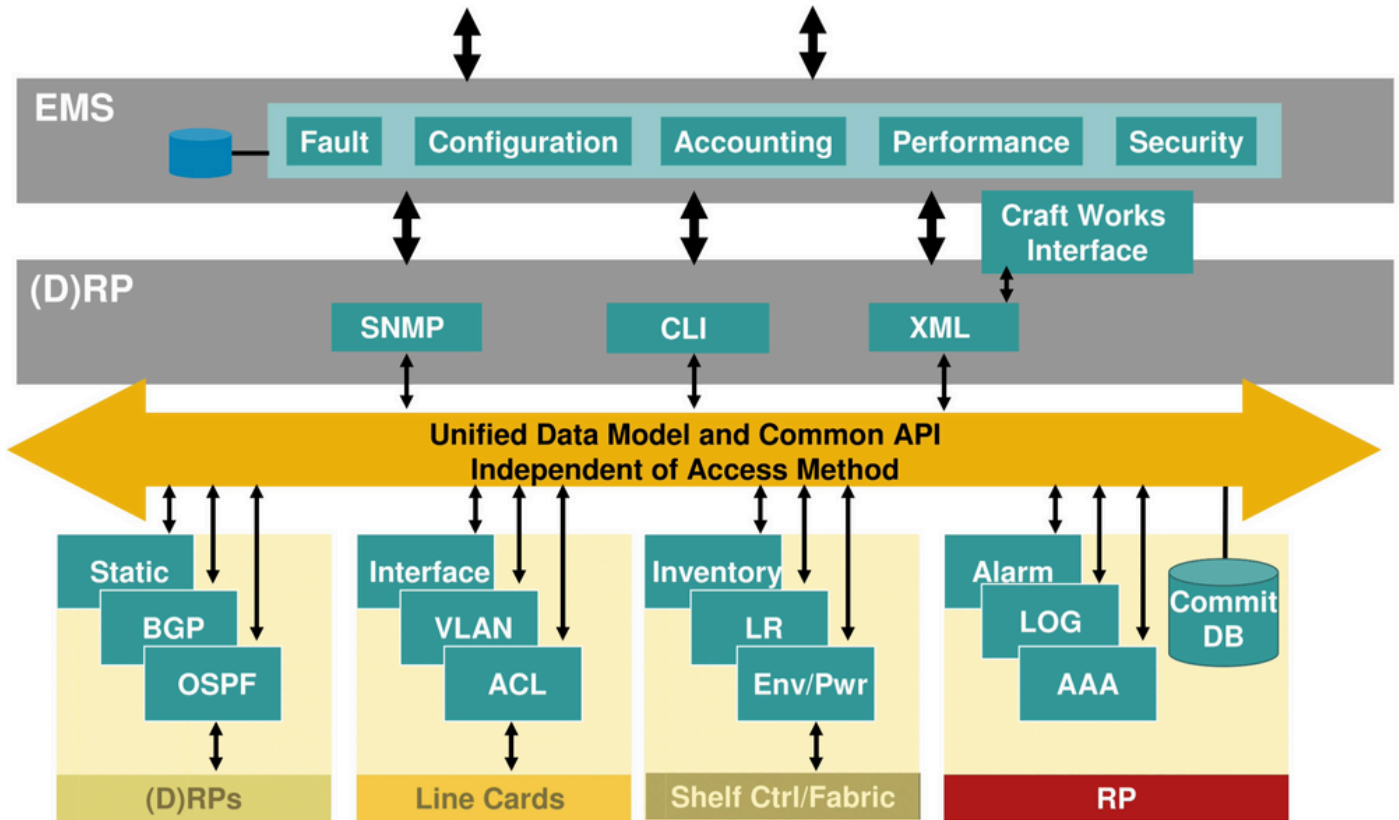
1. CORBA Agent (utilisant SSL).
2. Agent SSH.



Remarque : CORBA a été déconseillé après la version 3.7.

CORBA en tant que protocole hérité utilisait SSL, qui se trouve dans les packages de sécurité. D'autre part, la méthode SSH nécessite l'utilisation du package de gestion.

L'infra est distribuée de cette façon :



À partir de l'image précédente, le processus principal de XML est visible dans le processeur du routeur et du commutateur (RSP). Ces processus ont une API commune qui supervise toutes les informations des autres processus du périphérique, qui sont :

- Dans la carte de ligne :
 - Collecteur de statistiques
 - Serveur de statistiques
- À partir du processeur du routeur :
 - Gestionnaire de statistiques.
 - SysDB.
 - TCP (pour l'API XML, TCP 38751 est utilisé)
 - NetIO

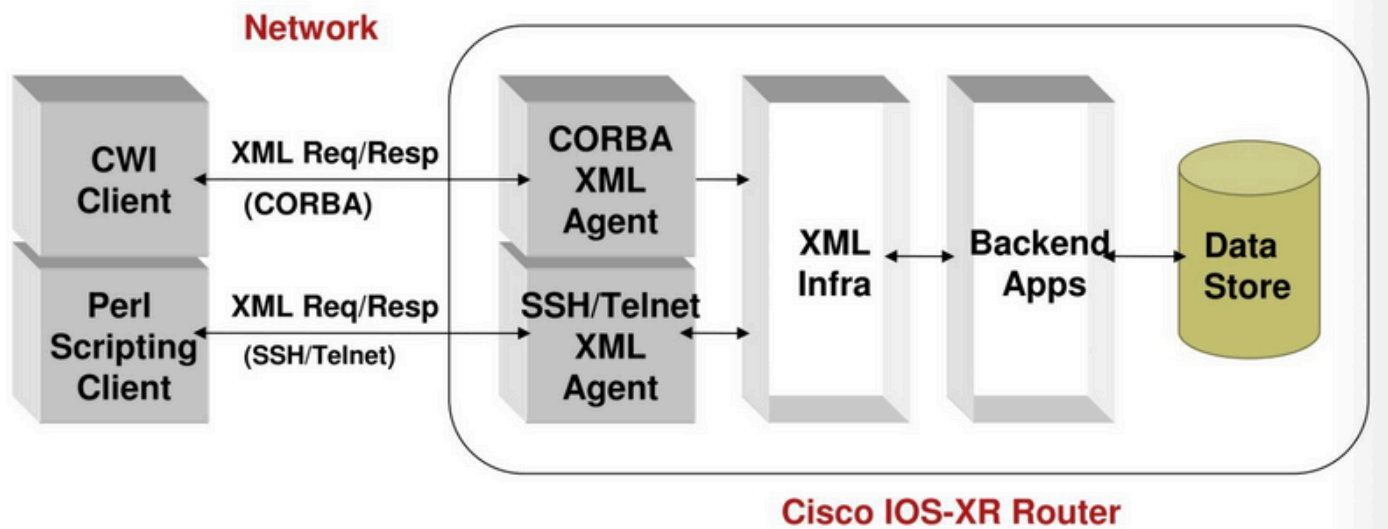


Remarque : pour plus d'informations à ce sujet, consultez la section Erreurs XML dans Cisco IOS XR.

Selon la demande, un ou plusieurs processus sont déclenchés. Lors du développement d'outils d'automatisation, si la réponse est supérieure à la taille du bloc, la réponse ne peut pas aboutir. Si un processus met du temps à répondre, il peut générer des journaux EDM, s'arrêter ou affecter des services.

Interaction API avec le client

L'image suivante illustre l'interaction :



Pour configurer/activer l'agent XML sur le périphérique, utilisez :

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

Le système doit être en accord avec le client dans les domaines suivants :

1. Versions
2. Schéma

Les versions peuvent être définies dans 2 zones :

- Balise de demande :

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

- Balise du composant principal :

Ce qui s'applique au composant spécifique :

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



Remarque : n'oubliez pas que vous pouvez vérifier la version du schéma en appliquant l'opération `GetVersionInfo` au conteneur.

-
- Mise à jour de version mineure : ajout au schéma XML, comme l'ajout d'un nouvel élément de données.
 - Mise à jour de version majeure : modifications sémantiques, suppressions du schéma ou du composant, etc.

Vous pouvez vérifier la version. L'exemple suivant montre comment procéder :



Remarque : les opérations XML sont GET, SET, etc.

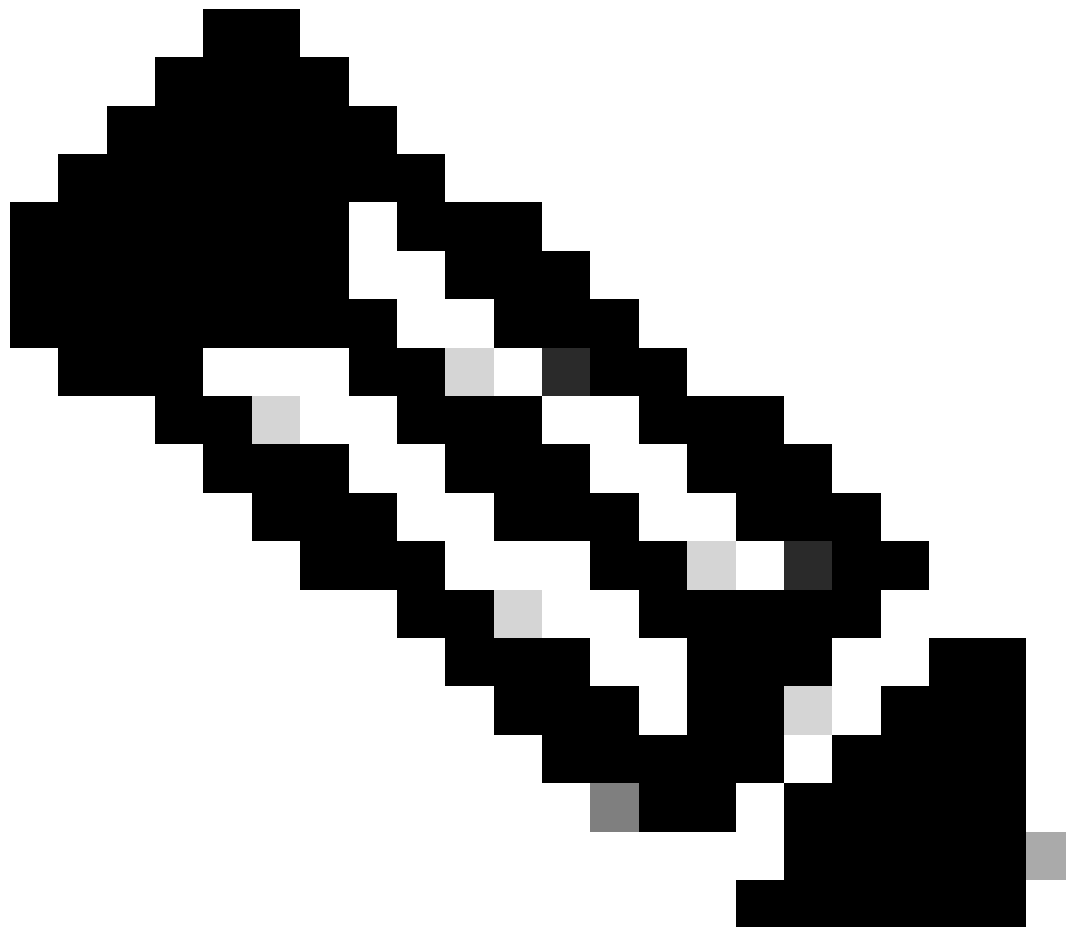
```
<?xml version="1.0" encoding="UTF-8"?>  
<Request MajorVersion="1" MinorVersion="0">  
<GetVersionInfo>  
<Operation/>  
</GetVersionInfo>  
</Request>
```

Utilisez la commande xml echo formal, puis ajoutez la balise GetVersionInfo à la requête pour trouver la version exécutée sur le routeur. comme illustré ici :

```
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>
```



Remarque : Request affiche toutes les versions en cours dans le composant principal et affiche également la version dans les conteneurs qu'il contient.

Erreurs XML dans Cisco IOS XR

Si le chemin est correct, chaque demande d'API XML affiche les informations demandées.

Lorsque la requête est incorrecte

Le routeur affiche trois messages différents :

- ÉlémentIntrouvable

Ce message s'affiche chaque fois qu'une opération GET a une réponse vide.

- ÉlémentIntrouvableCi-dessous

L'opération GET ne contient pas cette opération dans le schéma XML.

- Introuvable

L'élément demandé est introuvable au niveau de l'élément.

Types d'erreurs

1. Transport : les erreurs de cette catégorie incluent tout élément entre la communication agent/client XML. Cela signifie que toute interaction ou problème SSH peut se produire dans le transport. Par conséquent, pour vérifier ce genre de problèmes, il est conseillé de vérifier les traces SSH pour vérifier tout problème avec l'authentification, le port, etc.

2. XML Parser : Tout problème avec le format et la syntaxe, les problèmes dans la réponse envoyée ou dans la requête. Ces problèmes envoient généralement la raison de l'échec lorsqu'une erreur se produit.

Exemple :

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

3. Schéma XML : tout schéma différent du schéma du routeur. Pour résoudre ces problèmes, vérifiez le schéma et les versions de l'interface de ligne de commande.

Exemple :

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

4. Traitement de l'opération : Lorsque vous configurez le périphérique, ces problèmes peuvent se produire. Pour résoudre ces problèmes, vous devez dépanner les processus, par exemple commit, sysdb, etc.



Remarque : les informations d'erreur sont ajoutées au niveau de l'élément d'opération. Il est codé sous la forme d'attributs `ErrorCode` (int 32 bits) et `Errormsg`.

Autres recommandations

Autres techniques utiles :

1. Caractères génériques : il s'agit des requêtes spécifiques.
2. Traitement par lots : combinaison de plusieurs techniques ou opérations dans une seule demande (opérations au mieux).
3. Filtrage personnalisé : Si le schéma le permet, pour faciliter la sélection des lignes dans les tables.

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.