

Cisco IOS XR용 XML 트러블슈팅

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[메모리 조절 문제](#)

[추가 문제 설명](#)

[솔루션](#)

[Cisco IOS XR의 XML](#)

[XML API 사용](#)

[XML API에서 지원되는 작업](#)

[제공된 XML 스키마 서비스](#)

[쿼리 작성](#)

[XML 쿼리 테스트](#)

[Cisco IOS XR에서 XML 심층 분석](#)

[클라이언트와 API 상호 작용](#)

[Cisco IOS XR의 XML 오류](#)

[쿼리가 잘못된 경우](#)

[오류 유형](#)

[추가 권장 사항](#)

소개

이 문서에서는 서비스 요청 및 Cisco 툴에서 나타나는 XML 메모리 제한 문제에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Cisco IOS® XR 플랫폼
- XML(Extensible Markup Language) 인프라
- CORBA(Common Object Request Broker) 아키텍처
- EMS(Element Management System)
- 외부 데이터 관리자(EDM)

사용되는 구성 요소

이 문서의 정보는 ASR9000을 기반으로 합니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

메모리 조절 문제

다음과 같은 질문으로 케이스가 열렸습니다.

1. 라우터에서 구성할 수 있는 최대 메모리는 얼마입니까?
2. 대용량 XML 회신을 중단할 방법이 있습니까?

답변:

1. RSP/노드가 실행 중인 버전(cXR 및 eXR용 버전)에 따라 최대 메모리 변경

64비트 버전(eXR). 스로틀 메모리 크기가 하나 있습니다.

```
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)
```

32비트 버전(cXR)입니다. 또 다른 제한이 있습니다.

```
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.
```

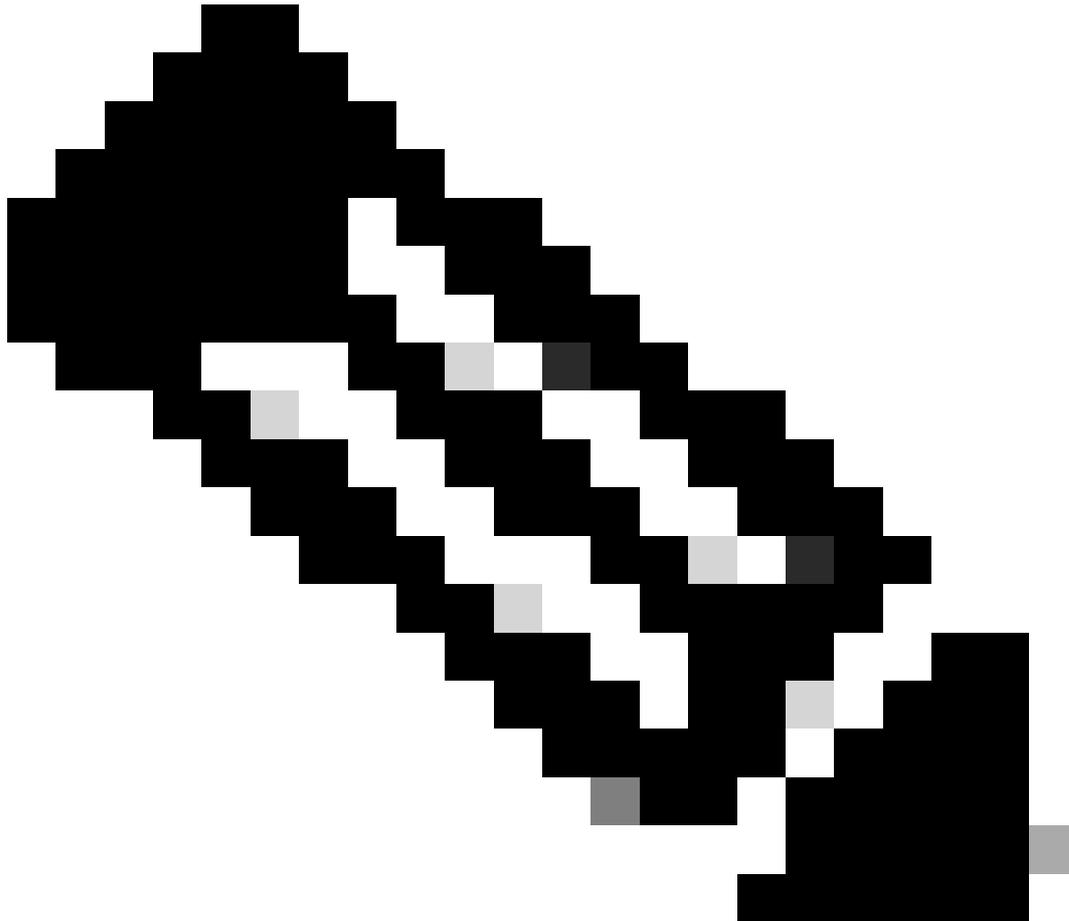
<snip>

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)



참고: 두 버전 모두 기본값은 300MB입니다

2. XML 회신을 끊는 방법은 여러 가지가 있습니다.

- 반복기:

반복기가 구성된 경우 XML 총 응답을 특정 창 크기로 분할합니다. 창에 반복기 크기가 있습니다. 예를 들어 응답이 1Gb이고 반복기 크기가 500Mb인 경우 XML을 2개의 회신으로 분할합니다.

이 메서드는 반복기 크기에 따라 GetNext 작업(응답에 필요한 작업)을 추가하여 응답을 변경합니다.

한 세션에 최대 10개의 반복기가 있을 수 있습니다.

- **조절(프로세스 속도):**

이 기능은 XML 프로세스에서 사용하는 메모리를 제한합니다. 프로세스가 메모리를 초과한 경우 "메모리 사용량에 대한 조절(throttle)에 도달했습니다"라는 오류 메시지가 표시됩니다.

- **스트리밍:**

스트리밍은 반복기와 마찬가지로 XML 응답을 특정 창으로 분할합니다. 차이점은 응답이며, GetNext 작업 및 반복자 ID를 제거합니다. XML은 스트리밍된 패키지를 전송하고 클라이언트는 스트림이 종료될 때 응답을 빌드합니다.

추가 문제 설명

자동화가 필요한 사용 사례의 경우, pyIOSXR 툴을 [사용합니다](#). 이 자동화 툴은 XML 에이전트이며, 일부 show 명령을 실행하고, 전체적으로 디바이스에 연결하는 데 도움이 됩니다.

이 상담원에게 큰 요청을 보낼 때마다 오류가 표시됩니다.

참고: 등록된 Cisco 사용자만 내부 Cisco 툴 및 정보에 액세스할 수 있습니다.

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

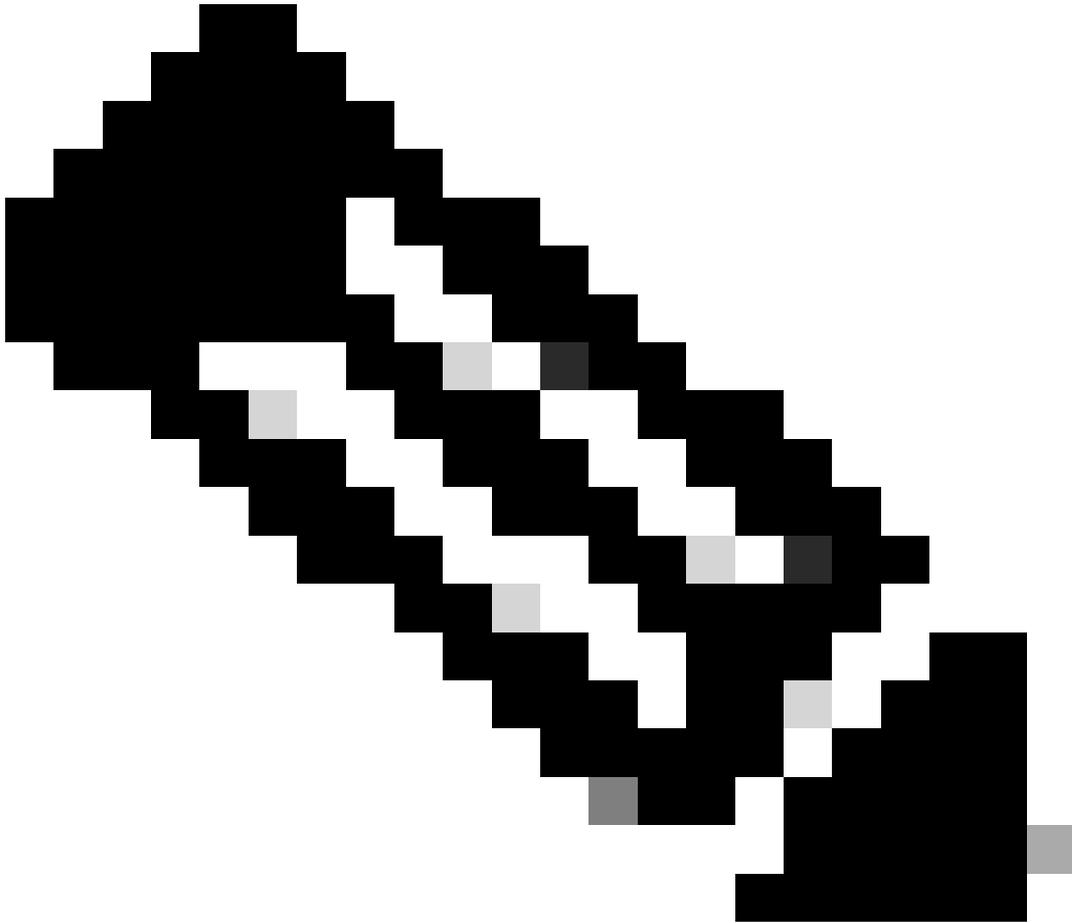
"더 작은 데이터를 쿼리하기 위한 요청 최적화"를 표시하는 오류 메시지와 함께 상담원 XML API를 반복기로 구성할 수 있습니다. 이는 이론적으로 응답을 세분화할 수 있습니다.

반복기가 구성되면 동일한 오류 메시지가 표시됩니다. "RESPONSE ERROR...Please optimize the request to query smaller data."

오류가 표시되면 다음 단계는 해당 기능(이 경우 반복기)이 질의에 대해 작동하지 않은 이유를 파악하는 것입니다.

[pyIOSXR](#)에서는 디바이스에서 XML 에이전트를 올바르게 활성화할 것을 권장합니다. 즉, API에서 반복기의 사용을 허용하지 않습니다.

다음 단계는 두 번째 옵션인 스트리밍을 테스트하는 것입니다.



참고: pyIOSXR에서는 다음 요소가 있는 기본 XML 쿼리 이외의 다른 헤더로 작업할 수 없습니다.

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

스트리밍 및 반복기는 쿼리에 헤더를 추가합니다. 스트리밍은 시스템이 전체 응답을 구축하는 데 도움이 되는 스트림 ID를 추가합니다. 반복기는 GET_Next 및 반복기 ID를 추가합니다.

스트리밍 테스트에서도 반복기와 동일한 오류 메시지가 나타났습니다.

솔루션

이 섹션에서는 장치에 대한 쿼리 추가 권장 사항을 참조하여 이러한 문제를 해결할 수 있습니다. 와일드카드 중 하나입니다. 와일드카드는 메모리 제한 제한에 대한 솔루션입니다.

와일드카드는 불필요한 정보를 요청하지 않도록 특정 쿼리를 빌드합니다. 예를 들어 BGP 정보의 경우 일반 show route 명령 대신 show route bgp 명령을 사용합니다. 이 예는 XML 쿼리 및 논리에 적용됩니다. 시스템에 대량 정보를 요청하면 메모리 및 처리 문제가 발생할 수 있습니다.

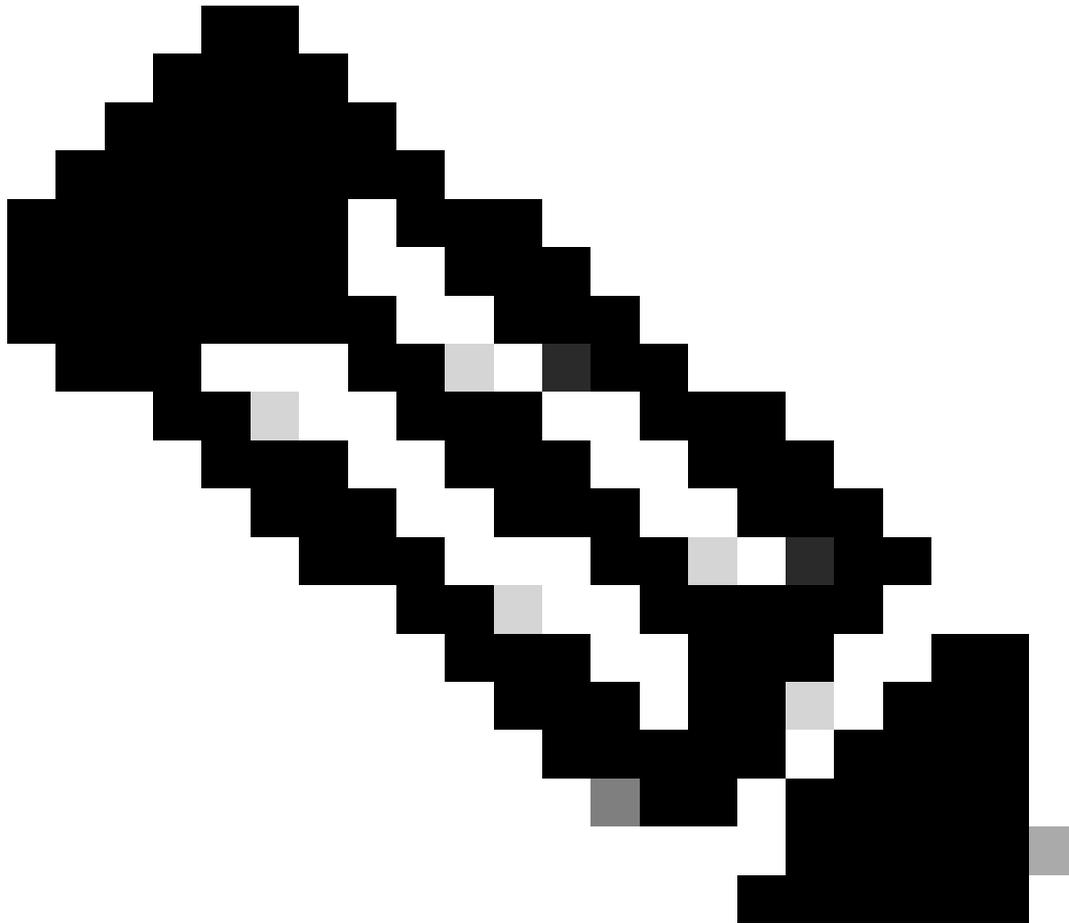
케이스를 열었을 때 이 쿼리가 사용되었습니다.

```
<?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>
                    </NetworkTable>
                  </AF>
                </AFTable>
              </VRF>
            </VRFTable>
          </InstanceActive>
        </Instance>
      </InstanceTable>
    </BGP>>
  </Operational>
</Get>
</Request>
```

이 쿼리는 전체 BGP 테이블을 대상으로 합니다. 이 비특정 요청의 경우 응답이 거의 2.2Gb였으므로 메모리 제한 제한에 도달했습니다.

이를 해결하려면 특정 쿼리가 필요하므로 시스템에서 쿼리를 처리하고 정보를 반환할 수 있습니다.

Cisco IOS XR의 XML



참고: 세부 정보, 사양 또는 기타 명령이 필요한 경우 다음 링크를 클릭하십시오. XML Programming guide: [XR XML programming guide](#)

XML은 데이터가 표시되고 구조화되는 방법을 정의합니다. 이는 컴퓨터가 비트로 이해하는 내용을 구문 분석하고, 정형화된 표준화된 정보를 표시하는 방법입니다.

XML의 구조는 다음과 같습니다.

```
<init>  
  <body>
```

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

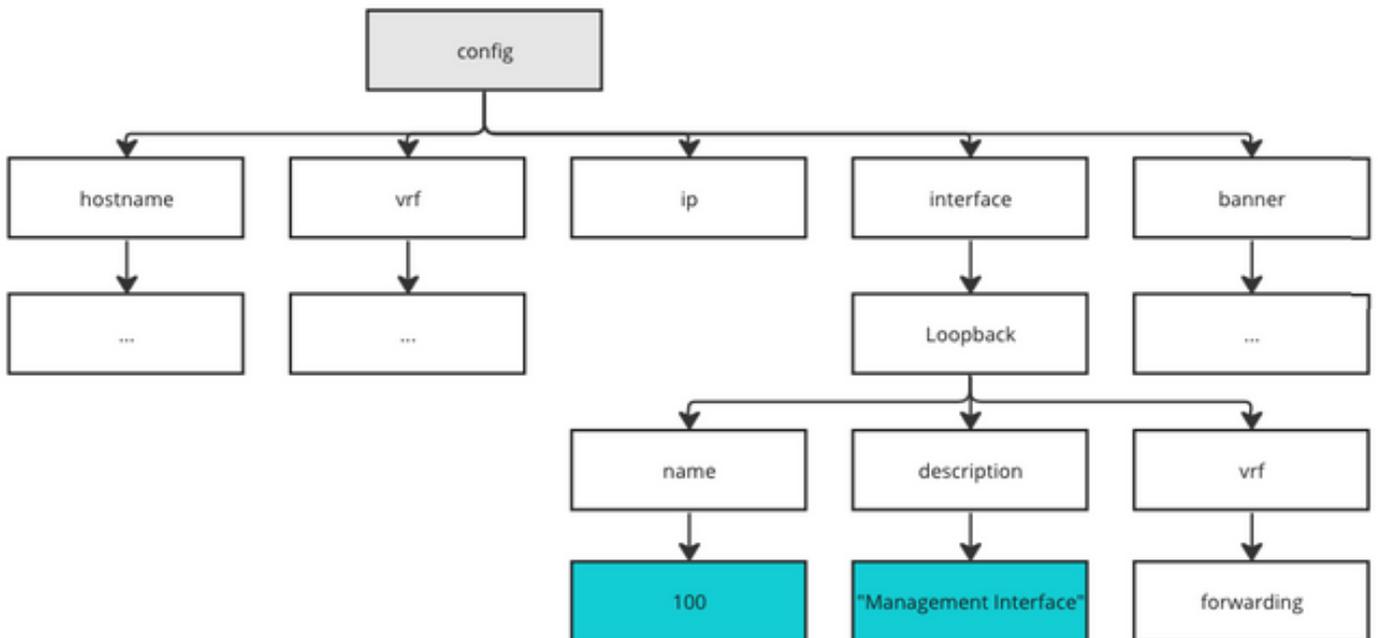
모든 태그는 두 부분으로 나뉘는데, 여는 태그(<init>)와 닫는 태그(</init>)입니다. 이 구조를 사용하지 않으면 태그가 끝나는 위치를 XML에서 인식할 수 없습니다.

XML은 두 개의 엔터티로 구성됩니다.

- 컨테이너
- 리프

이 구조화된 데이터에 대한 비유는 XML 데이터를 트리로 비교하는 것입니다. 용기는 나뭇가지이고 나뭇가지마다 잎을 박아 넣었습니다. Leaf에는 정보 이외의 다른 정보가 포함되지 않습니다.

예를 들어, 다음 트리는 뿌리가 회색이고, 컨테이너가 흰색이며, 잎이 파란색입니다.



XML API 사용

XML API를 테스트/활용하려면 먼저 쿼리가 필요합니다.

1. 질의에 헤더가 있습니다.

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



참고: 이 태그는 닫기 태그가 필요하지 않은 유일한 태그입니다.

2. 요청에 대한 태그를 추가해야 합니다. 요청에서 버전을 지정해야 합니다.

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

3. 헤더, 요청 및 버전의 경우 본문은 API에 포함된 모든 XML 작업으로 계속됩니다.

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



주: 요청에는 각 질의에 서로 다른 작업이 포함됩니다. 작업별로 요청을 수행할 필요는 없습니다.

다음 예에서는 모든 필수 정보가 포함된 요청을 표시합니다.

```
<?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 API에서 지원되는 작업

Cisco IOS XR은 사용자가 XML 스키마의 정보를 상호 작용할 수 있도록 하는 5가지 작업을 지원합니다.

1. 고유자료업무

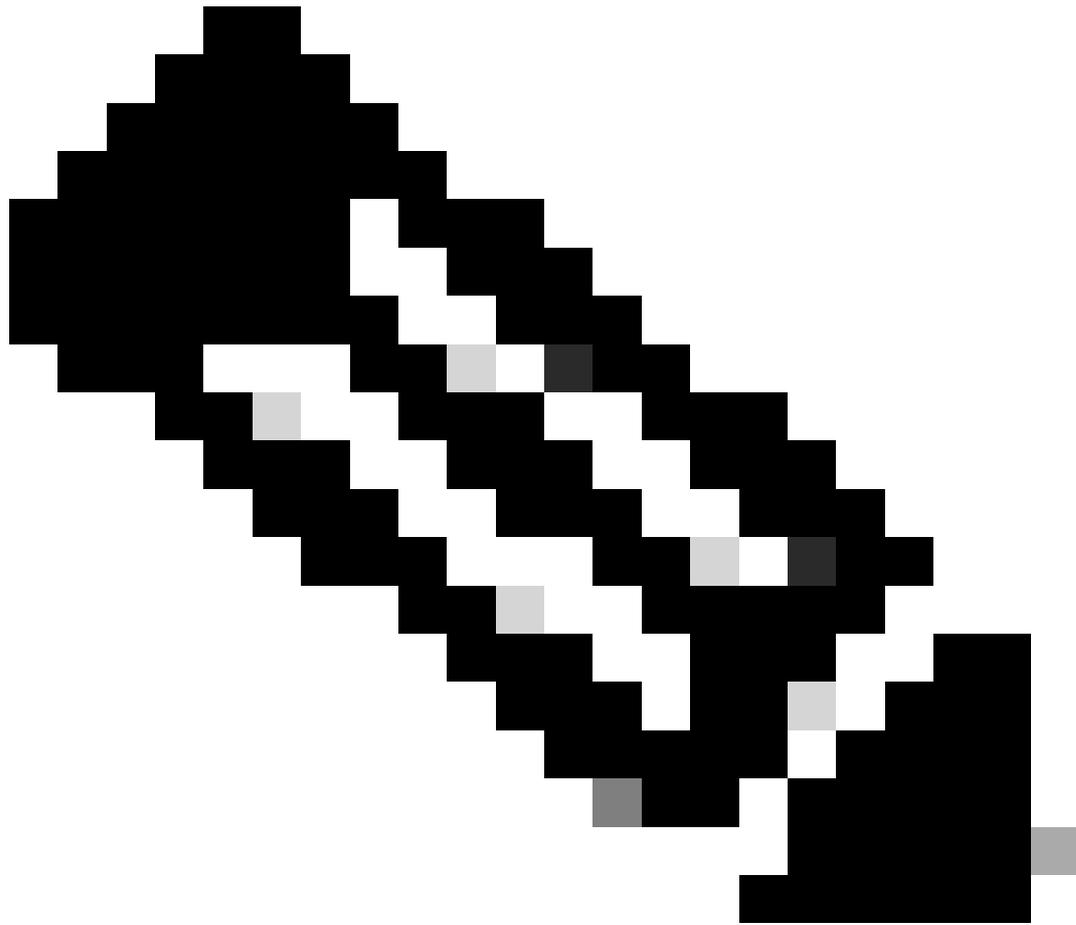
- `<Get></Get>`: 작업 데이터 항목(leaf에 있는 데이터)을 수집합니다.
- `<Set></Set>`: 데이터 항목을 구성, 변경 또는 추가합니다.
- `<Delete></Delete>`: 하나 이상의 데이터 항목을 제거합니다.
- `<GetVersionInfo></GetVersionInfo>`: 요청한 구성 요소의 주 버전 번호와 부 버전 번호를 검색하는 작업입니다.
- `<GetDataSpaceInfo></GetDataSpaceInfo>`: 이 작업 태그는 컨테이너에 매핑된 리프 이름을 표시합니다.

2. CLI 운영:

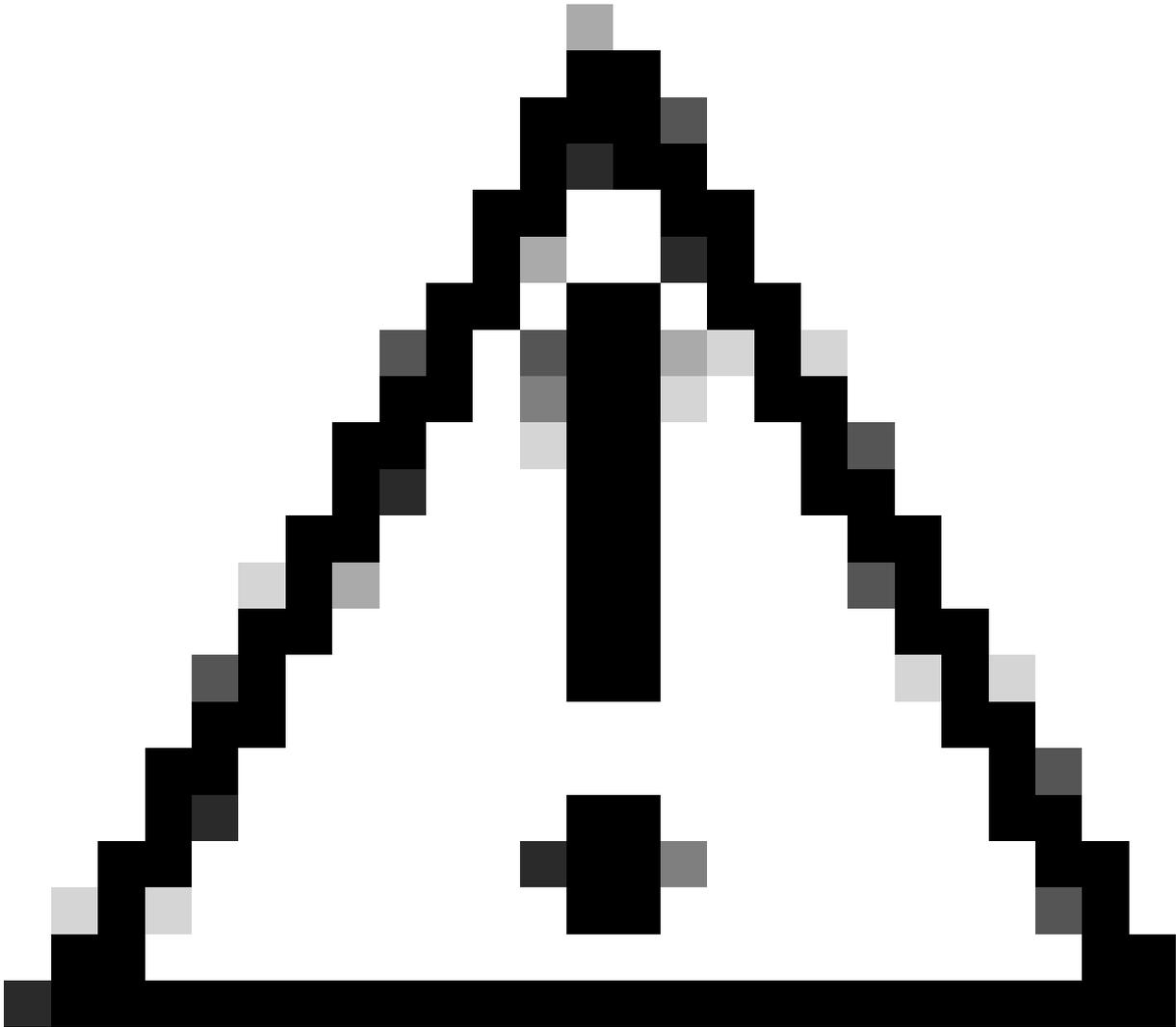
- `<CLI></CLI>`: 컨피그레이션 요청을 CLI 명령으로 실행할 수 있는 태그입니다.
- `<GetNext></GetNext>`: 데이터가 블록 크기보다 큰 경우 이 태그는 다음에 분할된 회신에 추가됩니다.

제공된 XML 스키마 서비스

XML API는 다음 네임스페이스를 구현합니다.



참고: 루트 개체를 확인해야 하는 경우 [Cisco IOS-XR XML Network Management를 참조하십시오.](#)



주의: 각 네임스페이스는 서로 다른 데이터 및 작업을 감독합니다.

-
- Configuration(컨피그레이션): Get, Set 및 Delete(가져오기, 설정 및 삭제) 작업을 허용하는 작업입니다.
 - 작동 가능: CLI show 명령과 같습니다.
 - 조치: 설정 조치를 지원합니다. 이 작업을 사용하면 clear 명령 및 기타 유사한 명령과 같은 작업 데이터에 액세스할 수 있습니다.
 - AdminOperational: 관리 운영 데이터에 액세스합니다.
 - AdminAction: 관리 작업 데이터에 액세스합니다.
 - AdminConfiguration: 관리 컨피그레이션 데이터에 액세스합니다.

쿼리 작성

여행에는 두 가지 다른 길이 있습니다. 쿼리에 대한 네임스페이스를 선택한 후에는 스키마를 사용하여 작업해야 합니다.

1. XML 스키마 설명서

이러한 옵션은 컨테이너와 leaf를 표시하는 전체 XML 트리를 제공합니다. 설명서를 보려면 [CRS XML Schemas\(CRS XML 스키마\)](#)를 클릭합니다.

설명서 페이지가 표시됩니다.

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
- <DHCPv6> - 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

이 페이지에는 객체, 컨테이너 및 1차 하위 구성요소가 표시됩니다. 각 개체에는 자식 개체가 포함되어 있습니다. 자식 개체에 다른 컨테이너가 없으면 leaf로 간주됩니다.

하위 항목을 클릭하면 쿼리를 빌드할 수 있는 정보가 표시됩니다. IPv4의 경우:

<IPV4>

Description: IPV4 related services

Task IDs required: ipv4

Parent objects:

[<Services>](#)

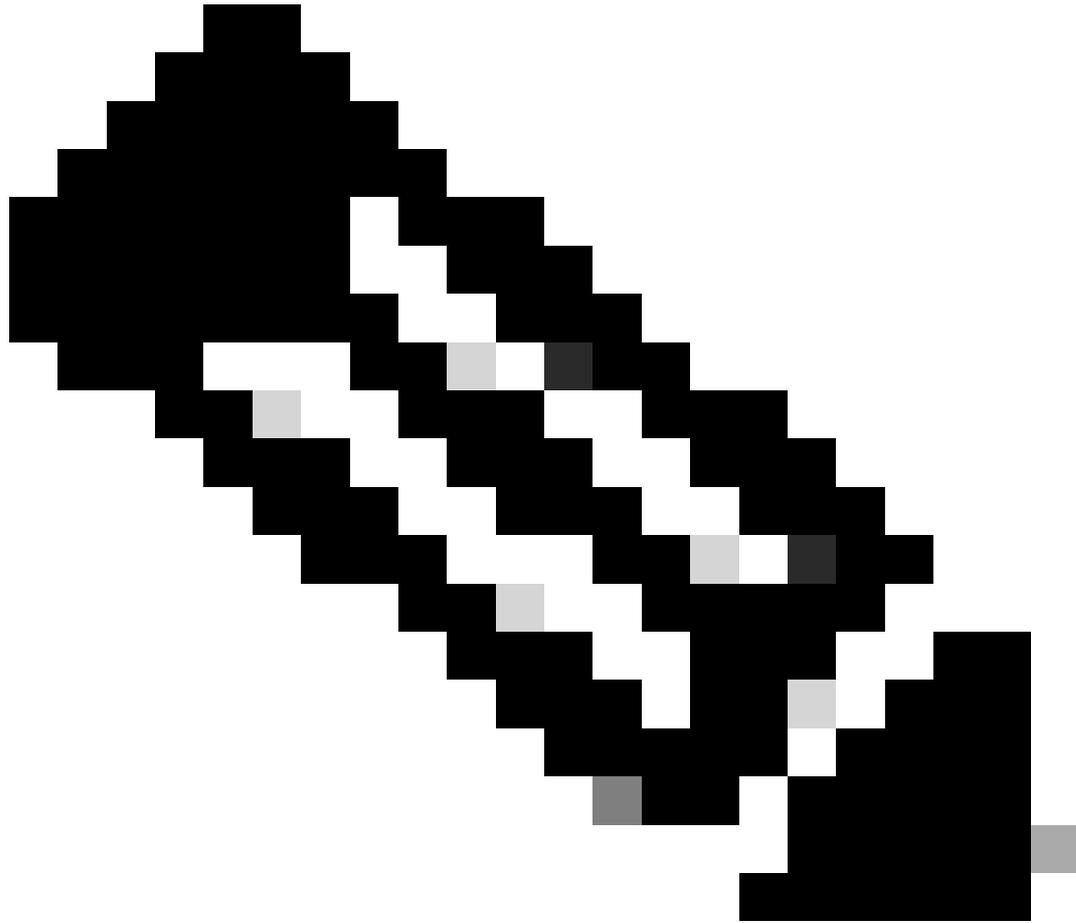
Child objects:

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

Available Paths:

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

이전 이미지에서 모든 IPv4 쿼리의 경우 컨테이너 IP, 컨테이너 Cinetd, 컨테이너 서비스, 리프 IPv4를 대상으로 컨피그레이션 작업을 실행할 수 있습니다.



참고: 쿼리를 수행하기 위해 leaf여야 하는 모든 작업입니다.

쿼리는 다음이 됩니다.

```
<?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 API는 Cisco IOS XR에 내장되어 있습니다. XML 트리를 보려면 show xml schema 명령을 사용합니다. 명령을 실행하면 전체 XML 스키마가 Linux와 유사한 CLI에서 표시됩니다.

이 CLI에는 다음 작업이 있습니다.

?:도움말

LS: 브랜치 사용자의 컨테이너/leaf 나열

<operation>: 명령을 실행하여 작업을 확인하고 명령을 실행할 수 있는지 확인할 수 있습니다.

일부 작업은 다음과 같습니다.

-설정

- adminconfig

- cd

- 목록

- 오퍼르

-작업

CD: 디렉터리/컨테이너를 변경합니다.

다음 예를 보여 줍니다.

```
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>
```



경고: 디바이스에 로그인하려면 자격 증명이 필요합니다. 이러한 자격 증명은 디바이스의 로컬이며 루트 ls/관리자 프로필이 필요합니다.

다음 예에서는 XML API를 사용하여 쿼리를 빌드하는 방법을 보여 줍니다. 예를 들어 쿼리는 XML 에이전트가 활성화되어 있는지 확인해야 합니다.

```
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
```

다음 프롬프트를 확인합니다.

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

이전 프롬프트에 XML 트리가 표시됩니다. 이 정보를 사용하여 쿼리는 다음과 같습니다.

```

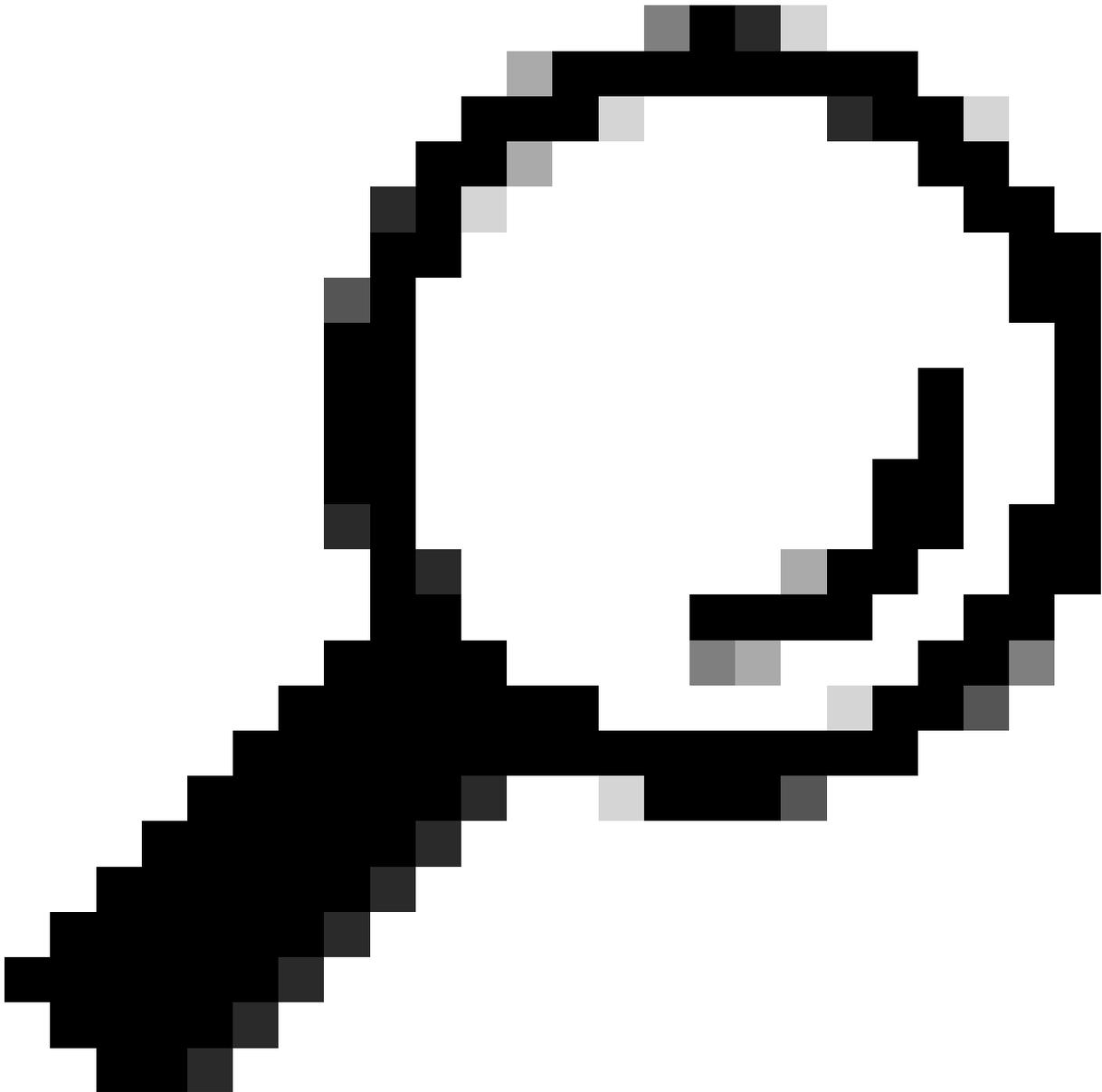
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">

```

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

XML 쿼리 테스트

쿼리가 빌드되면 다음 단계로 쿼리를 테스트합니다. 테스트하기 위해 CLI에서 `xml echo format` 명령을 실행할 수 있습니다. 이 작업은 동일한 디바이스에 보관할 수 있습니다.

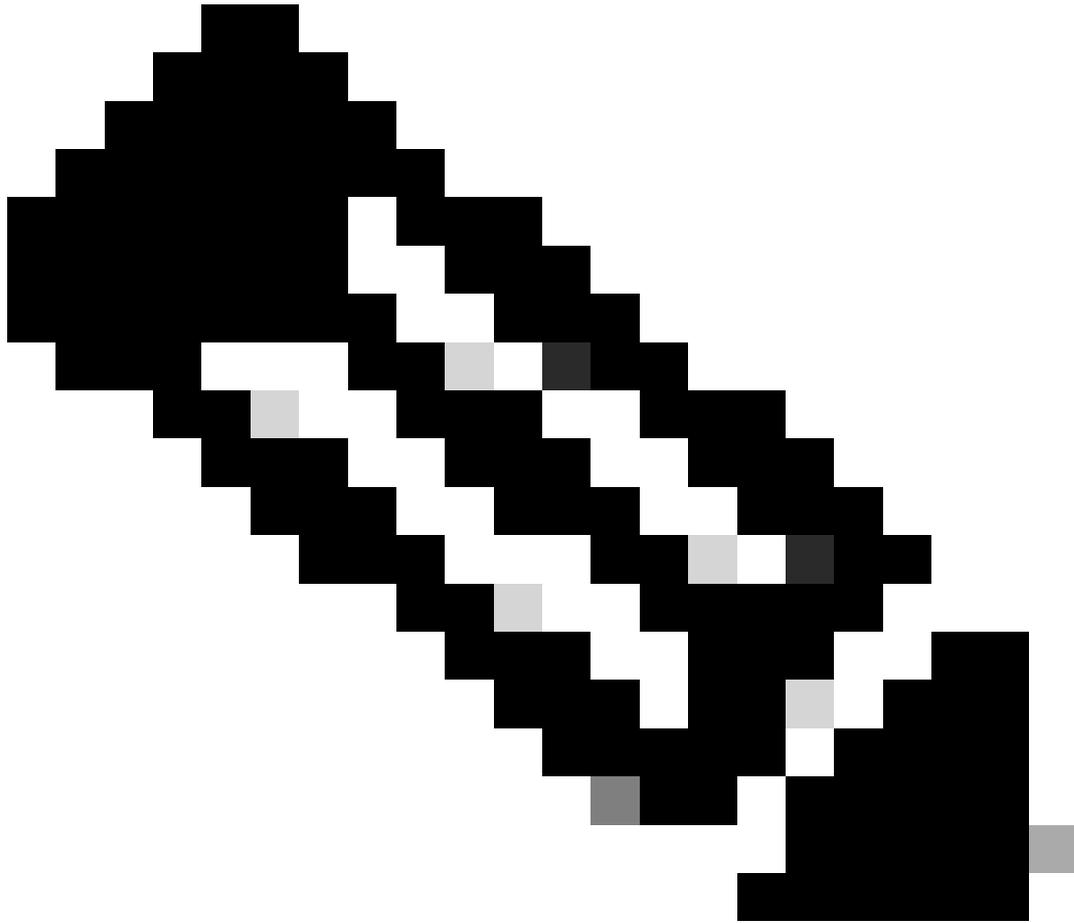


팁: 사용자가 이전 명령을 입력한 후 쿼리를 복사/붙여넣은 다음 enter 키를 누를 수 있습니다. 디바이스가 쿼리에 대한 응답을 표시합니다.

```
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
```

라우터에 응답 및 ResultSummary가 표시됩니다.



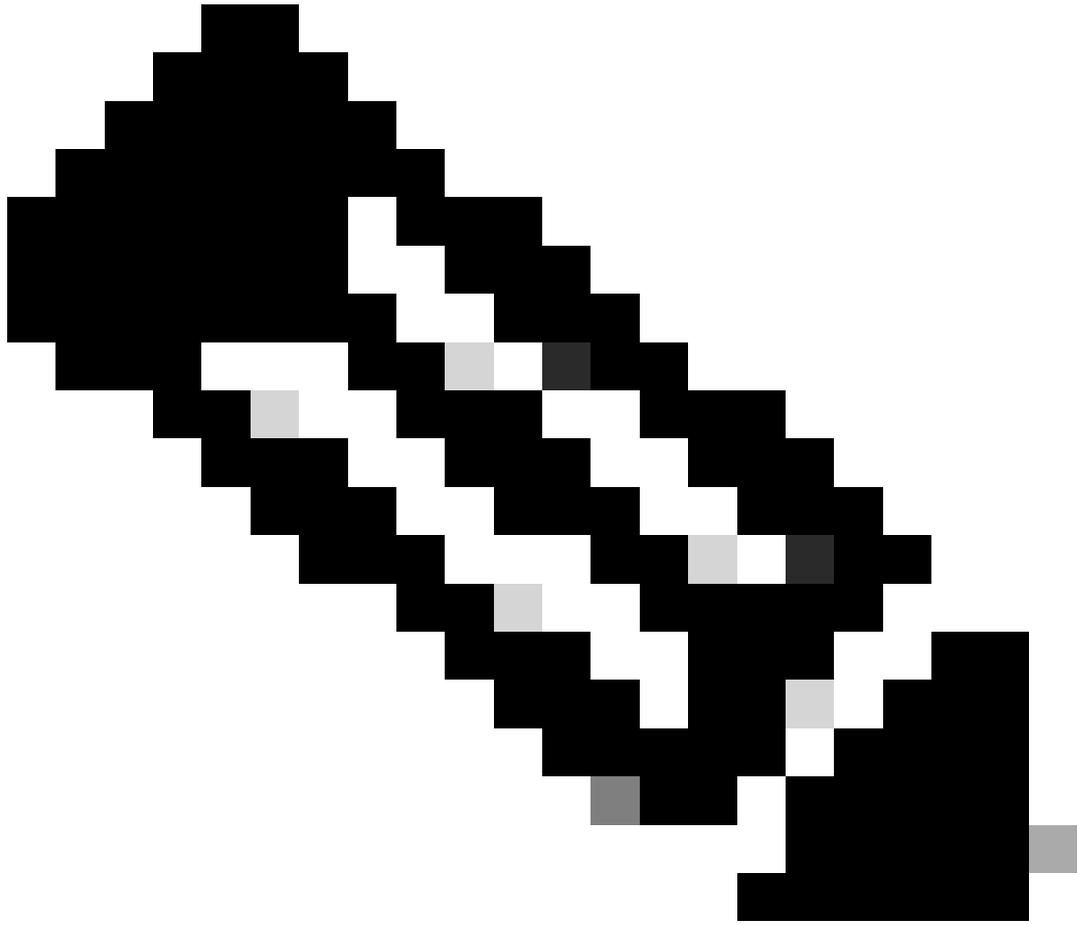
참고: 이 XML 응답은 표시되지 않으며 스토를 메모리를 구성할 필요가 없습니다. 어떤 SSH/TTY도 사용하지 않으므로, 크기에 관계없이 응답을 요청합니다. 한 가지는 XML 클라이언트와 다른 내장된 XML echo 명령이며, 동작은 변경될 수 있습니다.

Cisco IOS XR에서 XML 심층 분석

Cisco IOS XR은 서로 다른 패키지가 작동해야 하는 모듈형 시스템입니다. XML은 관리 평면 하위 시스템에 있는 것으로 간주됩니다.

XR가 시작되었을 때 디바이스에서 XML을 활용하는 두 가지 방법이 있었습니다.

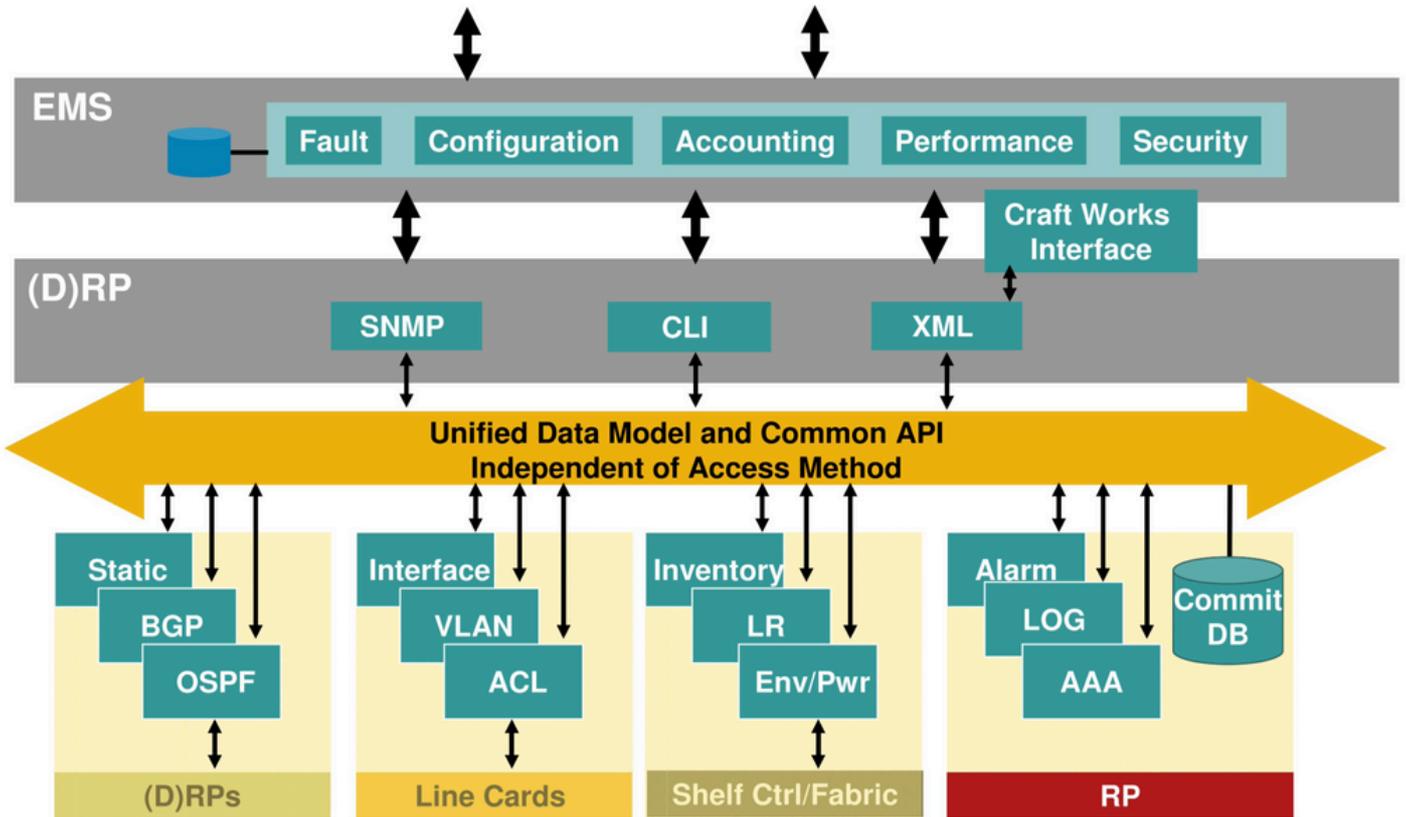
1. CORBA 에이전트(SSL 사용).
2. SSH 에이전트.



참고: CORBA는 버전 3.7 이후 더 이상 사용되지 않습니다.

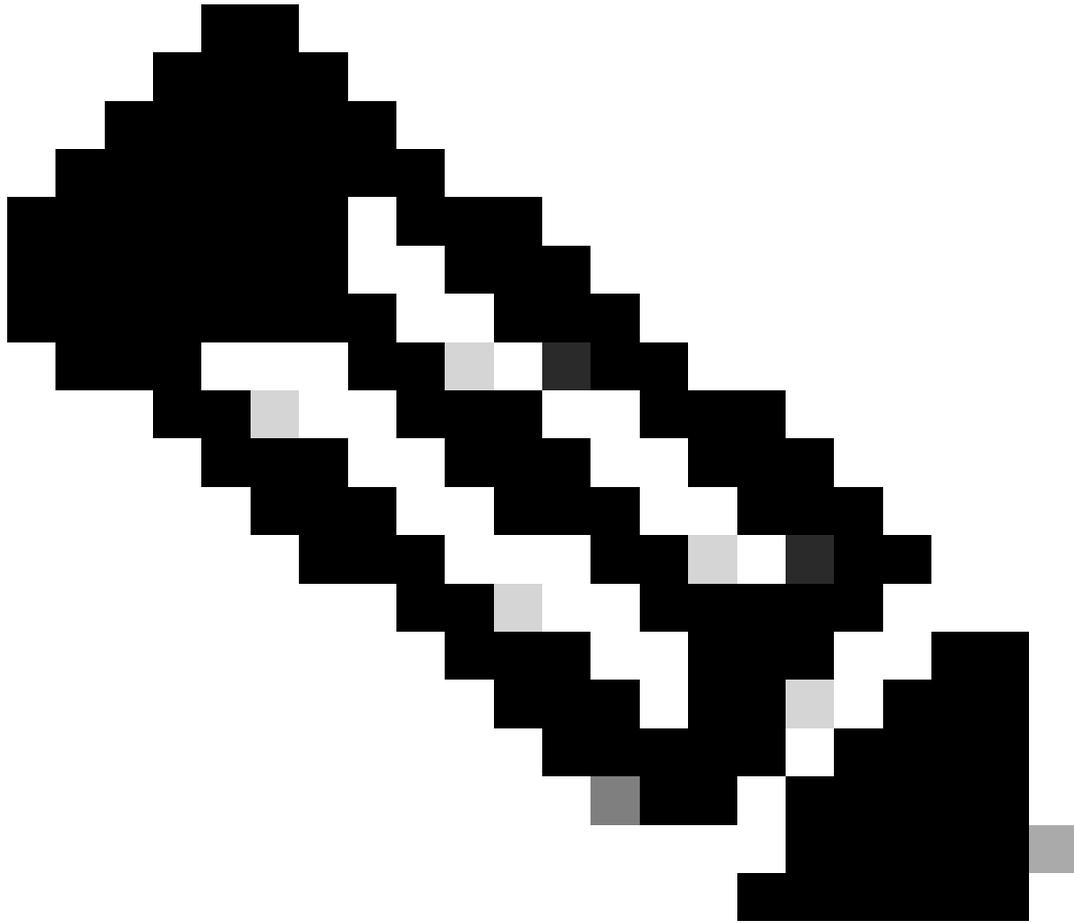
레거시 프로토콜인 CORBA는 보안 패키지에 있는 SSL을 사용했습니다. 반면, SSH 방법에는 관리 용이성 패키지를 사용해야 합니다.

인프라는 다음과 같이 분산됩니다.



이전 이미지에서 XML의 주요 프로세스는 RSP(Router and Switch Processor)에서 볼 수 있습니다. 이러한 프로세스에는 디바이스의 다른 프로세스에서 오는 모든 정보를 감독하는 공통 API가 있습니다.

- 라인 카드에서:
 - 통계 컬렉터
 - 통계 서버
- 라우터 프로세서에서:
 - 통계 관리자
 - SysDB
 - TCP(XML API의 경우 TCP 38751 사용)
 - Netlo

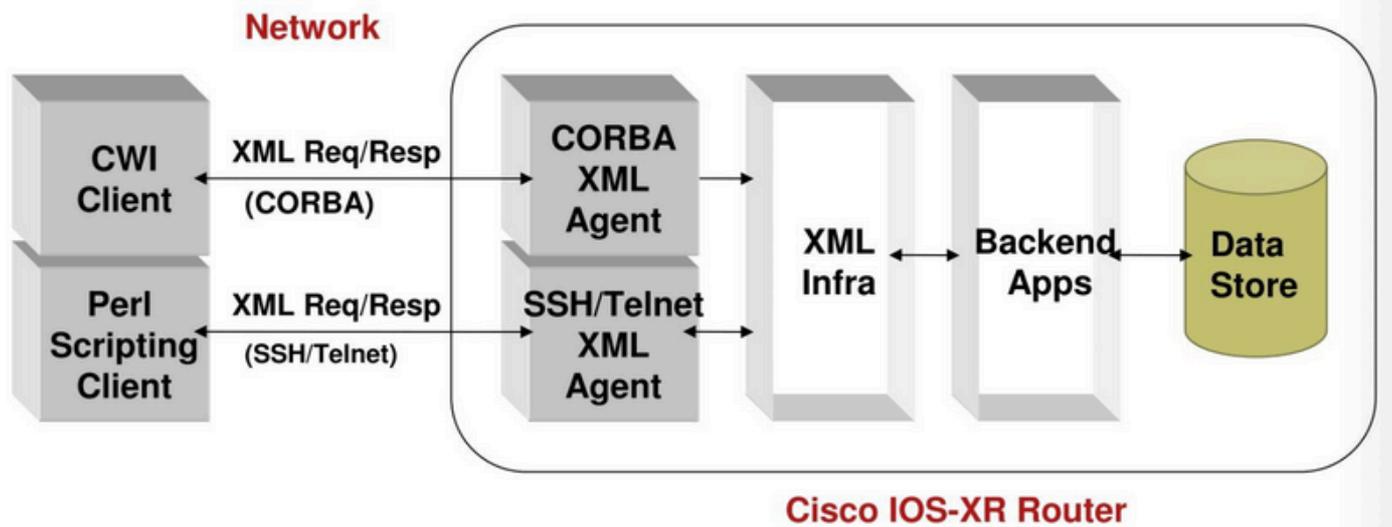


참고: 이에 대한 자세한 내용은 Cisco IOS XR 섹션의 XML 오류를 참조하십시오.

요청에 따라 하나 이상의 프로세스가 트리거됩니다. 자동화 도구를 개발할 때 응답이 블록 크기보다 크면 응답이 성공할 수 없습니다. 프로세스에 회신하는 데 시간이 오래 걸리는 경우 EDM 로그를 생성하거나, 종료하거나, 서비스에 영향을 줄 수 있습니다.

클라이언트와 API 상호 작용

다음 그림에서는 상호 작용을 보여줍니다.



디바이스에서 XML 에이전트를 구성/활성화하려면 다음을 사용합니다.

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

시스템은 다음 영역에서 클라이언트와 일치해야 합니다.

1. 버전
2. 스키마

버전은 2개의 영역으로 정의할 수 있습니다.

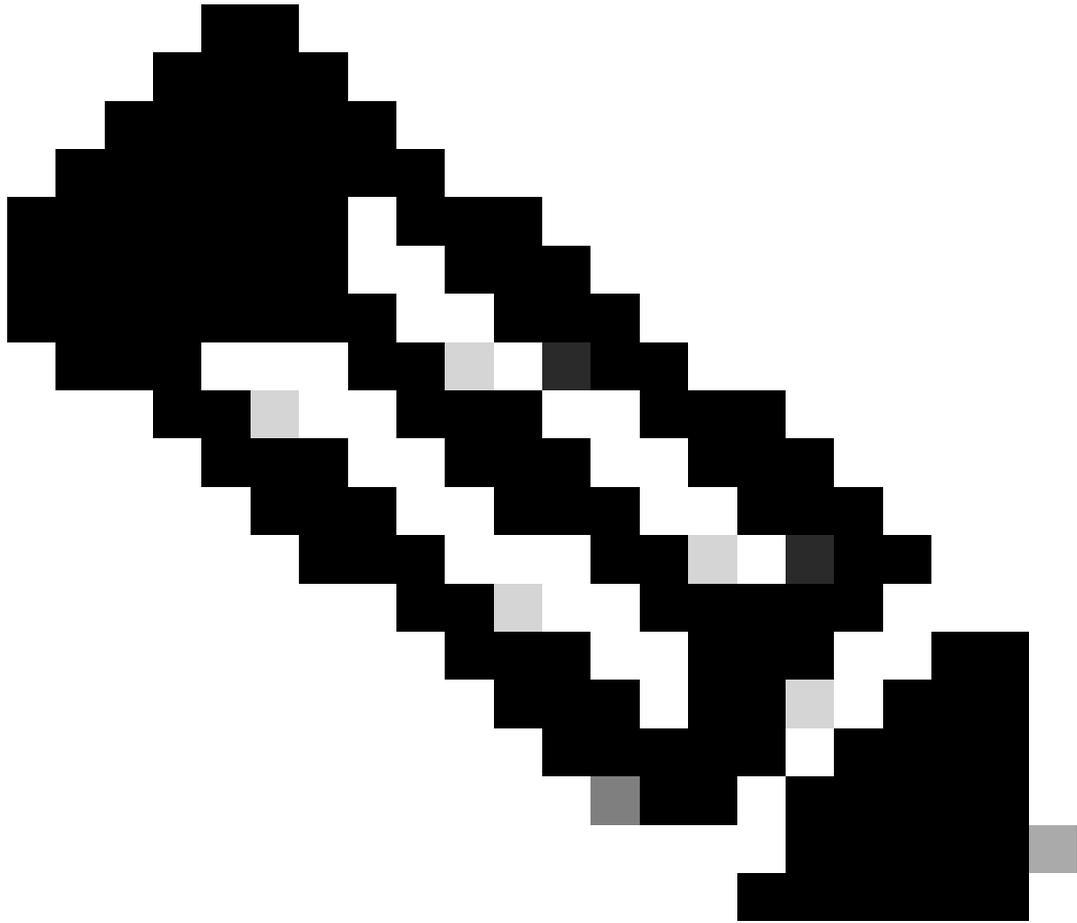
- 요청 태그:

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

- 주요 구성 요소 태그:

이는 특정 구성 요소에 적용됩니다.

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



참고: 컨테이너에 적용된 GetVersionInfo 작업으로 스키마 버전을 확인할 수 있습니다.

-
- 부 버전 업데이트: 새 데이터 항목을 추가하는 등 XML 스키마에 추가하는 모든 작업입니다.
 - 주 버전 업데이트: 의미 체계 변경, 스키마 또는 구성 요소에 대한 삭제 등

버전을 확인할 수 있습니다 다음 예에서는 이를 수행하는 방법을 보여 줍니다.

참고: XML 작업은 GET, SET 등입니다.

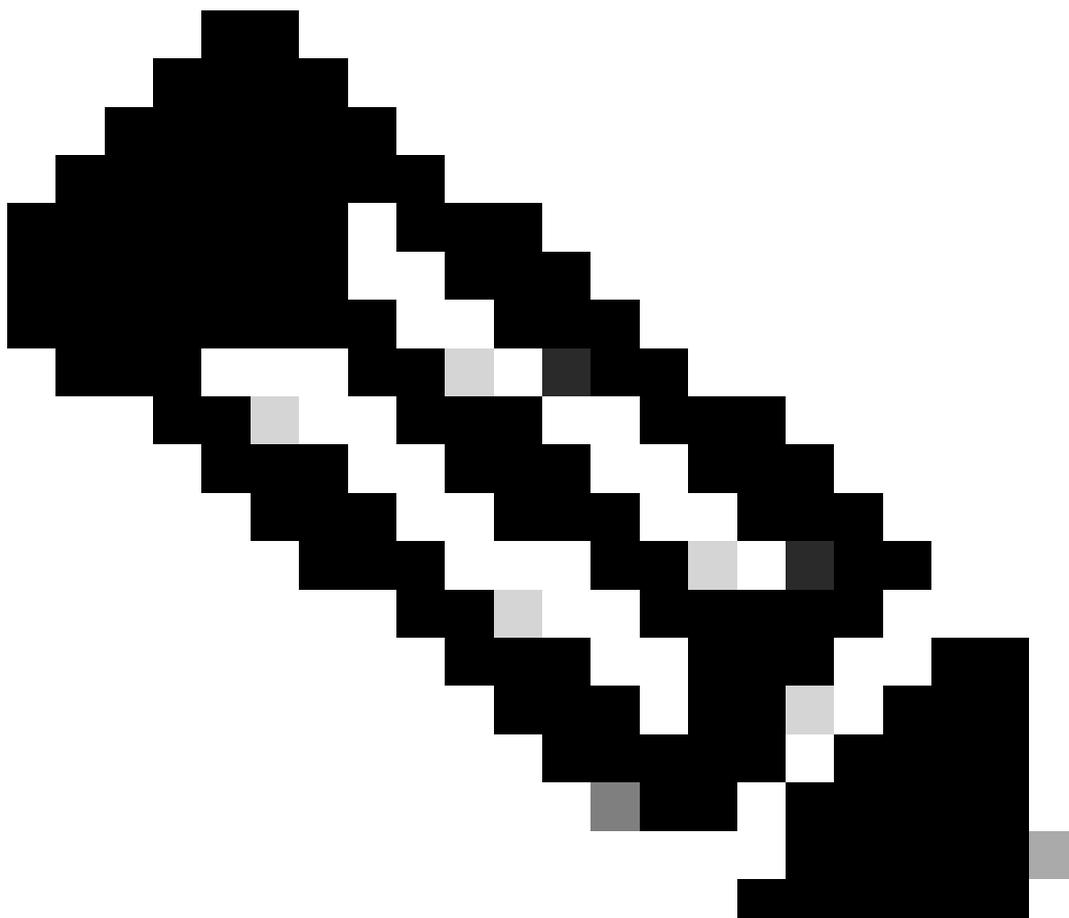
```
<?xml version="1.0" encoding="UTF-8"?>
<Request Major Version="1" MinorVersion="0">
<GetVersionInfo>
<Operation/>
</GetVersionInfo>
</Request>
```

xml echo formal 명령을 사용한 다음 GetVersionInfo 태그를 쿼리에 추가하여 라우터에서 실행 중인 버전을 찾습니다.

```
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>
```



주: 요청에는 주요 구성 요소에서 실행 중인 모든 버전이 표시되며, 해당 구성 요소 내의 컨테이너에도 버전이 표시됩니다.

Cisco IOS XR의 XML 오류

경로가 올바르면 각 XML API 요청에서 요청된 정보를 표시합니다.

쿼리가 잘못된 경우

라우터에 세 가지 메시지가 표시됩니다.

- 항목을 찾을 수 없음

이 메시지는 GET 작업에 빈 응답이 있을 때마다 표시됩니다.

- 아래 항목 없음

GET 작업에 XML 스키마에 이 작업이 없습니다.

- 찾을 수 없음

요청한 요소가 요소 수준을 찾을 수 없습니다.

오류 유형

1. Transport(전송): 이 범주의 오류는 XML 에이전트/클라이언트 통신 사이의 모든 것을 포함합니다. 이는 모든 SSH 상호 작용 또는 문제가 전송에서 발생할 수 있음을 의미합니다. 따라서 이러한 종류의 문제를 확인하려면 SSH 추적을 검사하여 인증, 포트 등과 관련된 문제를 확인하는 것이 좋습니다.

2. XML 파서: 형식 및 구문에 대한 모든 문제, 전송 응답 또는 쿼리에서 문제. 이러한 문제는 일반적으로 오류가 발생할 때 실패 사유를 보냅니다.

예를 들면 다음과 같습니다.

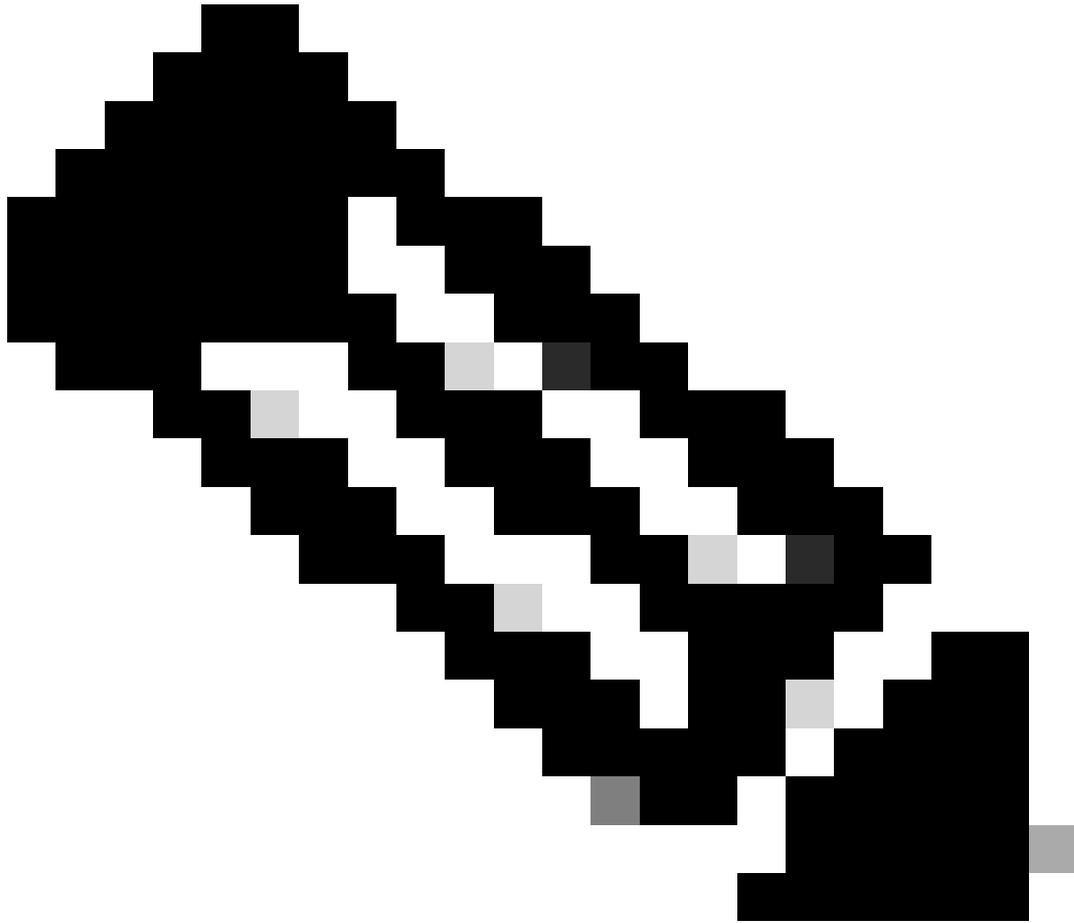
```
ERROR: 0xa367a600 'XML Service Library' detected the 'fatal' condition 'The throttle on the memory usage
```

3. XML 스키마: 라우터 스키마와 다른 모든 스키마 이러한 문제를 해결하려면 CLI 스키마 및 버전을 확인합니다.

예를 들면 다음과 같습니다.

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

4. 작업 처리: 장치를 구성할 때 이러한 문제가 발생할 수 있습니다. 이 문제를 해결하려면 커밋, sysdb 등과 같은 프로세스 트러블슈팅이 필요합니다.



주: 오류 정보는 공정 요소 레벨에 추가됩니다. 이 코드는 ErrorCode(32비트 int) 및 Errormsg 특성의 형태로 코딩됩니다.

추가 권장 사항

기타 유용한 기술:

1. 와일드카드: 특정 쿼리라고도 합니다.
2. 일괄 처리: 하나의 요청에 여러 가지 기술이나 작업을 결합하는 것(최선의 작업)
3. 사용자 정의 필터링: 스키마에서 허용하는 경우 테이블에서 행을 선택할 수 있습니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.