



## CHAPTER 9

# Cisco XML Schema Versioning

---

Before the router can carry out a client application request, it must verify version compatibility between the client request and router component versions.

Major and minor version numbers are included on the <Request> and <Response> elements to indicate the overall XML application programming interface (API) version in use by the client application and router. In addition, each component XML schema exposed through the XML API has a major and minor version number associated with it.

This chapter describes the format of the version information exchanged between the client application and the router, and how the router uses this information at run time to check version compatibility.

This chapter contains these sections:

- [Major and Minor Version Numbers, page 9-107](#)
- [Run-Time Use of Version Information, page 9-108](#)
- [Retrieving Version Information, page 9-113](#)
- [Retrieving Schema Detail, page 9-115](#)

## Major and Minor Version Numbers

The top-level or root object (that is, element) in each component XML schema carries the major and minor version numbers for that schema. A minor version change is defined as an addition to the XML schema. All other changes, including deletions and semantic changes, are considered major version changes.

The version numbers are documented in the header comment contained in the XML schema file. They are also available as <xsd:appinfo> annotations included as part of the complex type definition for the top-level schema element. This enables you to programmatically extract the version numbers from the XML schema file to include in XML request instances sent to the router. The version numbers are carried in the XML instances using the MajorVersion and MinorVersion attributes.

This example shows the relevant portion of the complex type definition for an element that carries version information:

```
<xsd:complexType name="ipv4_bgp_cfg_BGP_type">
  <xsd:annotation>
    <xsd:documentation>BGP Configuration Commands</xsd:documentation>
  <xsd:appinfo>
    <ObjectType>Container</ObjectType>
    <MajorVersion>24</MajorVersion>
    <MinorVersion>0</MinorVersion>
    <TaskIdInfo TaskGrouping="Single">
```

```

        <TaskName>bgp</TaskName>
    </TaskIdInfo>
    <Parents>
        <Parent>
            <Schema>native_data_operations</Schema>
            <Name>Configuration</Name>
        </Parent>
    </Parents>
</xsd:appinfo>
</xsd:annotation>
    .
    .
    .
    <xsd:attributeGroup ref="VersionAttributeGroup"/>
    .
    .
    ..
</xsd:complexType>

```

The attribute group `VersionAttributeGroup` is defined as:

```

<xsd:attributeGroup name="VersionAttributeGroup">
    <xsd:annotation>
        <xsd:documentation>
            Common version information attributes
        </xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="MajorVersion" type="xsd:unsignedInt" use="required"/>
    <xsd:attribute name="MinorVersion" type="xsd:unsignedInt" use="required"/>
</xsd:attributeGroup>

```

## Run-Time Use of Version Information

Each XML request must contain the major and minor version numbers of the client at the appropriate locations in the XML. These version numbers are compared to the version numbers running on the router.

The behavior of the router, whether the request is accepted or rejected, depends on the value set for the `AllowVersionMismatch` attribute.

All requests are accepted when the `AllowVersionMismatch` attribute is set as `TRUE`. The request is then accepted or rejected based on these rules when the `AllowVersionMismatch` attribute is set as `FALSE`:

- If there is a major version discrepancy, then the request fails.
- If there is a minor version lag, that is, the client minor version is behind that of the router, then the request is attempted.
- If there is a minor version creep, that is, the client minor version is ahead of that of the router, then the request fails.
- If the version information has not been included in the request, then the request fails.
- The default value is used when the request does not specify the `AllowVersionMismatch` attribute. The default value is currently set as `TRUE`.

Each XML response can also contain the version numbers at the appropriate locations in the XML.

**Note**

If the client minor version is behind that of the router, then the response may contain elements that are not recognized by the client application. The client application must be able to handle these additional elements.

## Placement of Version Information

This example shows the placement of the MajorVersion and MinorVersion attributes within a client request to retrieve the global BGP configuration data for a specified autonomous system:

### Sample Client Request Showing Placement of Version Information

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get>
    <Configuration>
      <BGP MajorVersion="24" MinorVersion="0">
        <AS>
          <Naming><AS>0</AS></Naming>
          <FourByteAS>
            <Naming><AS>3</AS></Naming>
            <DefaultVRF/>
          </FourByteAS>
        </AS>
      </BGP>
    </Configuration>
  </Get>
</Request>
```

### Sample XML Response from the Router

```
<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0">
  <Get>
    <Configuration>
      <BGP MajorVersion="24" MinorVersion="0">
        <AS>
          <Naming><AS>0</AS></Naming>
          <FourByteAS>
            <Naming><AS>3</AS></Naming>
            <DefaultVRF>
              ...
              data returned here
              ...
            </DefaultVRF>
          </FourByteAS>
        </AS>
      </BGP>
    </Configuration>
  <Get>
    <ResultSummary ErrorCount="0"/>
</Response>
```

## Version Lag with the AllowVersionMismatch Attribute Set as TRUE

The example shows a request and response with a version mismatch. In this case, because the `AllowVersionMismatch` attribute is set as `TRUE`, the request is attempted. This is also the default behavior when `AllowVersionMismatch` attribute is not specified in the request. The router attempts the request and if the request is successful returns a `VersionMismatchExists` attribute at the appropriate point within the response along with a `VersionMismatchExistsBelow` attribute on the `<Get>` operation tag.



### Note

The version number, which is returned in the response, is the version running on the router. The versions in this example are hypothetical.

### Sample XML Client Request with a Version Mismatch

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get AllowVersionMismatch="true">
    <Configuration>
      <BGP MajorVersion="24" MinorVersion="0">
        <AS>
          <Naming><AS>0</AS></Naming>
          <FourByteAS>
            <Naming><AS>3</AS></Naming>
            <DefaultVRF>
              <Global/>
            </DefaultVRF>
          </FourByteAS>
        </AS>
      </BGP>
    </Configuration>
  </Get>
</Request>
```

### Sample XML Response from the Router

```
<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0" IteratorID="1">
  <Get AllowVersionMismatch="true" VersionMismatchExistsBelow="true">
    <Configuration>
      <BGP MajorVersion="24"
        MinorVersion="1">
        VersionMismatchExists="true">
      <AS>
        <Naming><AS>0</AS></Naming>
        <FourByteAS>
          <Naming><AS>3</AS></Naming>
          <DefaultVRF>
            <Global>
              ...
              data returned here
              ...
            </Global>
          </DefaultVRF>
        </FourByteAS>
      </AS>
    </BGP>
  </Configuration>
</Get>
<ResultSummary ErrorCount="0" />
</Response>
```

## Version Lag with the AllowVersionMismatch Attribute Set as FALSE

The example shows a request and response with a version mismatch, but the request specifies the AllowVersionMismatch attribute as FALSE.

In this case, the client minor version is behind the router, so the request is still attempted, but VersionMismatchExists and VersionMismatchExistsBelow attributes are not returned in the response.



### Note

The version number returned in the response is the version number running on the router. The versions in this example are hypothetical.

### Sample XML Client Request with the AllowVersionMismatch Attribute Set as False

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get AllowVersionMismatch="false">
    <Configuration>
      <BGP MajorVersion="24" MinorVersion="0">
        <AS>
          <Naming><AS>0</AS></Naming>
          <FourByteAS>
            <Naming><AS>3</AS></Naming>
            <DefaultVRF>
              <Global/>
            </DefaultVRF>
          </FourByteAS>
        </AS>
      </BGP>
    </Configuration>
  </Get>
</Request>
```

### Sample XML Response from the Router

```
<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0" IteratorID="1">
  <Get AllowVersionMismatch="false">
    <Configuration>
      <BGP MajorVersion="24"
        MinorVersion="1">
        <AS>
          <Naming><AS>0</AS></Naming>
          <FourByteAS>
            <Naming><AS>3</AS></Naming>
            <DefaultVRF>
              <Global>
                ...
                data returned here
                ...
              </Global>
            </DefaultVRF>
          </FourByteAS>
        </AS>
      </BGP>
    </Configuration>
  </Get>
  <ResultSummary ErrorCount="0"/>
</Response>
```

## Version Creep with the AllowVersionMismatch Attribute Set as TRUE

The example shows a request and response with a version mismatch. In this case, the client is the AllowVersionMismatch attribute and is set as TRUE. The request is attempted.



### Note

The version number returned in the response is the version number running on the router. The versions in this example are hypothetical.

### Sample XML Request with an AllowVersion Mismatch Attribute Set as TRUE

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get AllowVersionMismatch="true">
    <Configuration>
      <BGP MajorVersion="24" MinorVersion="1">
        <AS>
          <Naming><AS>0</AS></Naming>
          <FourByteAS>
            <Naming><AS>3</AS></Naming>
            <DefaultVRF>
              <Global/>
            </DefaultVRF>
          </FourByteAS>
        </AS>
      </BGP>
    </Configuration>
  </Get>
</Request>
```

### Sample XML Response from the Router

```
<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0" IteratorID="1">
  <Get AllowVersionMismatch="true" VersionMismatchExistsBelow="true">
    <Configuration>
      <BGP MajorVersion="24"
        MinorVersion="0">
        VersionMismatchExists="true">
        <AS>
          <Naming><AS>0</AS></Naming>
          <FourByteAS>
            <Naming><AS>3</AS></Naming>
            <DefaultVRF>
              <Global>
                ...
                data returned here
                ...
              </Global>
            </DefaultVRF>
          </FourByteAS>
        </AS>
      </BGP>
    </Configuration>
  </Get>
  <ResultSummary ErrorCount="0" />
</Response>
```

## Version Creep with the AllowVersionMismatch Attribute Set as FALSE

The example shows a request and response with a version mismatch. In this case, the client minor version is ahead of the router minor version, which results in an error response.

### Sample XML Request with an AllowVersion Mismatch Attribute Set as FALSE

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get AllowVersionMismatch="false">
    <Configuration>
      <BGP MajorVersion="24" MinorVersion="1">
    </Configuration>
  </Get>
</Request>
```

### Sample XML Response from the Router

```
<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0" IteratorID="12345678">
  <Get ErrorCode="0x43679000">
    ErrorMessage="&apos;XML Service Library&apos; detected the &apos;warning&apos; condition &apos;An error was encountered in the XML beneath this operation tag&apos;";" >
  <Configuration>
    <BGP MajorVersion="24" MinorVersion="0" ErrorCode="0x4368ac00"
      ErrorMessage="&apos;XMLMDA&apos; detected the &apos;warning&apos; condition &apos;The XML version specified in the XML request is not compatible with the version running on the router&apos;"/>
  </Configuration>
</Get>
<ResultSummary ErrorCount="0"/>
</Response>
```

## Retrieving Version Information

The version of the XML schemas running on the router can be retrieved using the <GetVersionInfo> tag followed by the appropriate tags identifying the names of the desired components.

In this example, the <GetVersionInfo> tag is used to retrieve the major and minor version numbers for the BGP component configuration schema:

### Sample XML Request to Retrieve Major and Minor Version Numbers

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

### Sample XML Response from the Router

```
<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0">
  <GetVersionInfo>
    <Configuration>
      <BGP MajorVersion="18" MinorVersion="0"/>
    </Configuration>
```

```

    </GetVersionInfo>
    <ResultSummary ErrorCount="0" />
</Response>

```

This example shows how to retrieve the version information for all configuration schemas available on the router:

### Sample XML Request to Retrieve Version Information for All Configuration Schemas

```

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

```

### Sample XML Response from the Router

```

<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0">
  <GetVersionInfo>
    <Configuration>
      <AAA MajorVersion="2" MinorVersion="0">
        <ServerGroups MajorVersion="2" MinorVersion="3">
          <RADIUSServerGroupTable MajorVersion="2" MinorVersion="0"/>
          <TACACSServerGroupTable MajorVersion="5" MinorVersion="1"/>
        </ServerGroups>
        <TaskgroupTable MajorVersion="2" MinorVersion="3"/>
        <UsergroupTable MajorVersion="2" MinorVersion="3"/>
        <UsernameTable MajorVersion="2" MinorVersion="3"/>
        <DefaultTaskgroup MajorVersion="2" MinorVersion="3"/>
        <RADIUS MajorVersion="2" MinorVersion="0"/>
        <TACACS MajorVersion="5" MinorVersion="1"/>
      </AAA>
      . . . .
      <PPP MajorVersion="1" MinorVersion="0">
        <FSM MajorVersion="1" MinorVersion="0"/>
        <Authentication MajorVersion="1" MinorVersion="0"/>
        <CHAP MajorVersion="1" MinorVersion="0"/>
        <MS-CHAP MajorVersion="1" MinorVersion="0"/>
        <PAP MajorVersion="1" MinorVersion="0"/>
      </PPP>
      . . . .
      <QOS MajorVersion="4" MinorVersion="0"/>
      <IntfVRF MajorVersion="1" MinorVersion="1"/>
      <SONET MajorVersion="3" MinorVersion="0"/>
      <BGP MajorVersion="18" MinorVersion="0"/>
      <OSPF MajorVersion="7" MinorVersion="0"/>
      <IPV4_ACLAndPrefixList MajorVersion="5" MinorVersion="0"/>
      <IPV4Network MajorVersion="4" MinorVersion="0"/>
      <IPV4NetworkGlobal MajorVersion="4" MinorVersion="0"/>
      <NTP MajorVersion="2" MinorVersion="1"/>
      <ISIS MajorVersion="15" MinorVersion="0"/>
      <VRFTable MajorVersion="1" MinorVersion="1">
        <VRF><IPARM MajorVersion="3" MinorVersion="0"/>
          <AFI_SAFI_Table>
            <AFI_SAFI>
              <BGP MajorVersion="18" MinorVersion="0"/>
            </AFI_SAFI>
          </AFI_SAFI_Table>
        </VRF>
      </VRFTable>
      . . .
    </Configuration>

```



```

</GetVersionInfo>
<ResultSummary ErrorCount="0"/>
</Response>

```

## Retrieving Schema Detail

The SchemaDetail boolean attribute can now be specified on the <GetVersionInfo> operation to instruct the router to return additional schema detail in the response. If the SchemaDetail attribute is specified in the request, each schema entity in the <GetVersionInfo> response contains three additional boolean attributes listed in [Table 9-1](#).

**Table 9-1** Content Attributes

Content Attribute	Description
ContainsNaming	Indicates whether or not the schema entity contains naming information.
Getable	Indicates whether or not <Get> operations are supported for this schema.
Setable	Indicates whether or not <Set> operations are supported for this schema.

This example shows a request and response with the SchemaDetail attribute:

### Sample XML Client Request for Schema Detail

```

<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <GetVersionInfo SchemaDetail="true">
    <Configuration/>
  </GetVersionInfo>
</Request>

```

### Sample XML Response from the Router

```

<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0">
  <GetVersionInfo SchemaDetail="true">
    <Configuration/>
    ...
    <BGP MajorVersion="10" MinorVersion="1" ContainsNaming="false" Getable="true"
      Setable="true" Supported="true"/>
    <LACP MajorVersion="1" MinorVersion="0" ContainsNaming="false" Getable="true"
      Setable="true" Supported="true"/>
    <CDP MajorVersion="1" MinorVersion="1" ContainsNaming="false" Getable="true"
      Setable="true" Supported="true"/>
    <LR MajorVersion="0" MinorVersion="0" ContainsNaming="false" Getable="true"
      Setable="true" Supported="false"/>
    <InterfaceConfigurationTable MajorVersion="2" MinorVersion="0"
      ContainsNaming="false" Getable="true" Setable="true" Supported="true">
      <InterfaceConfiguration ContainsNaming="true" Getable="true" Setable="true">
        <Bundle MajorVersion="1" MinorVersion="0" ContainsNaming="false" Getable="true"
          Setable="true" Supported="true"/>
        <LACP MajorVersion="1" MinorVersion="0" ContainsNaming="false" Getable="true"
          Setable="true" Supported="true"/>
      </InterfaceConfiguration>
    </InterfaceConfigurationTable>
    ...
  </GetVersionInfo>
</Response>

```

```
        </InterfaceConfiguration>
    </InterfaceConfigurationTable>
    ...
</Configuration>
</GetVersionInfo>
<ResultSummary ErrorCount="0" />
</Response>
```