



XML Manageability

XML Manageability in Cisco IOS XR Software enables XML agent services for parsing, generating, and validating XML documents. It supports programmatically working with XML, allows custom functions for XML tags, and validates document types using DTD. The XML API provides a programmable interface for router configuration and monitoring, accessible via TCP, SSL, or VRF instances. Configuration involves enabling XML agents, specifying request method (TCP or SSL), setting iteration size and idle timeout, configuring processing capabilities, and enabling VRF for communication.

- [XML Manageability Overview, on page 1](#)

XML Manageability Overview

You can enable Extensible Markup Language (XML) agent services on the routers running Cisco IOS XR Software. The XML Parser Infrastructure provides parsing and generation of XML documents with Document Object Model (DOM), Simple API for XML (SAX), and Document Type Definition (DTD) validation capabilities:

- DOM allows customers to programmatically create, manipulate, and generate XML documents.
- SAX supports user-defined functions for XML tags.
- DTD allows for validation of defined document types.

The Cisco IOS XR XML API provides a programmable interface to the router by external management applications. This interface provides a mechanism for router configuration and monitoring using XML-formatted request and response streams. The XML interface is built on top of the Management Data API (MDA), which provides a mechanism for Cisco IOS XR components to publish their data models through MDA schema definition files.

Cisco IOS XR software provides the ability to access the router via XML using a dedicated TCP connection, Secure Socket Layer (SSL), or a specific VPN routing and forwarding (VRF) instance.

Configure XML Manageability

To configure XML manageability, you must configure an XML SSL agent to communicate through VRFs.

Step 1

Enable the XML requests over a dedicated TCP connection or a Secure Socket Layer (SSL) for the dedicated XML agent, using the **xml agent** command. You can also configure the *iteration on size*, *session timeout*, and *throttle memory*.

Example:

```
Router:# config
Router(config):# xml agent ssl
Router(config-xml-agent)# iteration on size 500
Router(config-xml-agent)# session timeout 5
Router(config-xml-agent)# throttle memory 300
Router(config-xml-agent)# commit
```

Similarly, you can use the **xml agent tcp** command to enable the XML requests over a dedicated TCP connection."

Step 2

In this example, we've configured the XML agent to receive and send messages via VRF1 and VRF2.

Example:

```
Router# config
Router(config)# xml agent
Router(config-xml-agent)# vrf VRF1
Router(config-xml-agent)# vrf VRF2
Router(config-xml-agent)# commit
```
