



YANG data models for programmability features

Cisco IOS XR supports a programmatic way of configuring and collecting operational data of a network device using YANG data models. Although configurations using CLIs are easier and human-readable, automating the configuration using model-driven programmability results in scalability.

The data models are available in the release image, and are also published in the [Github](#) repository. Navigate to the release folder of interest to view the list of supported data models and their definitions. Each data model defines a complete and cohesive model, or augments an existing data model with additional XPath. To view a comprehensive list of the data models supported in a release, navigate to the *Available-Content.md* file in the repository.

You can also view the data model definitions using the [YANG Data Models Navigator](#) tool. This GUI-based and easy-to-use tool helps you explore the nuances of the data model and view the dependencies between various containers in the model. You can view the list of models supported across Cisco IOS XR releases and platforms, locate a specific model, view the containers and their respective lists, leaves, and leaf lists presented visually in a tree structure. This visual tree form helps you get insights into nodes that can help you automate your network.

To get started with using the data models, see the *Programmability Configuration Guide*.

- [Access data models, on page 1](#)
- [Get started with IOS XR YANG data models, on page 4](#)

Access data models

You can access the data models using one of these options:

Access data models from router

To access data models directly from the router, you can use these steps:

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Router#configure
```

Step 2 Configure the NETCONF network management protocol to remotely configure and manage the router using YANG data models.

Example:

```
Router(config)#netconf-yang agent ssh
```

Step 3 Commit the configuration.

Example:

```
Router(config)#commit
```

Step 4 Establish a NETCONF session with the device and retrieve the capabilities information.

Example:

```
Router#show netconf-yang capabilities
Tue Sep 19 22:03:26.305 UTC
[Netconf capabilities]
```

```
D: Has deviations
```

Capability	Revision	D
urn:ietf:params:netconf:base:1.1	-	
urn:ietf:params:netconf:capability:candidate:1.0	-	
urn:ietf:params:netconf:capability:confirmed-commit:1.1	-	
urn:ietf:params:netconf:capability:interleave:1.0	-	
urn:ietf:params:netconf:capability:notification:1.0	-	
urn:ietf:params:netconf:capability:rollback-on-error:1.0	-	
urn:ietf:params:netconf:capability:validate:1.1	-	
http://cisco.com/ns/yang/Cisco-IOS-XR-8000-fib-platform-cfg	2019-04-05	
http://cisco.com/ns/yang/Cisco-IOS-XR-8000-lpts-oper	2022-05-05	
http://cisco.com/ns/yang/Cisco-IOS-XR-8000-platforms-npu-resources-oper	2020-10-07	
http://cisco.com/ns/yang/Cisco-IOS-XR-8000-qos-oper	2021-06-28	
http://cisco.com/ns/yang/Cisco-IOS-XR-Ethernet-SPAN-act	2021-03-22	
http://cisco.com/ns/yang/Cisco-IOS-XR-Ethernet-SPAN-cfg	2022-07-13	
http://cisco.com/ns/yang/Cisco-IOS-XR-Ethernet-SPAN-datatypes	2021-10-06	
http://cisco.com/ns/yang/Cisco-IOS-XR-Ethernet-SPAN-oper	2022-09-05	
http://cisco.com/ns/yang/Cisco-IOS-XR-aaa-aaacore-cfg	2019-04-05	
http://cisco.com/ns/yang/Cisco-IOS-XR-aaa-ldapd-cfg	2022-06-22	
http://cisco.com/ns/yang/Cisco-IOS-XR-aaa-ldapd-oper	2022-05-20	
http://cisco.com/ns/yang/Cisco-IOS-XR-aaa-lib-cfg	2020-10-22	
http://cisco.com/ns/yang/Cisco-IOS-XR-aaa-lib-datatypes		

----- Truncated for brevity -----

By examining the capabilities, you can view the available data models for the software version installed on the router.

Access data models from Cisco Feature Navigator

To access data models from Cisco Feature Navigator, you can use these steps:

Procedure

Step 1 Go to [Cisco Feature Navigator](#).

- Step 2** If you have a Cisco.com account, click on the **Login** button and enter your credentials. If you don't have an account, you can click **Continue as Guest**.
- You will be directed to the Cisco Feature Navigator main page.
- Step 3** Click **YANG Data Models**.
- Step 4** Select the **Product** and **Cisco IOS XR Release** based on your requirement.
- The data models are listed based on type—Cisco XR native models, Unified models and OpenConfig models.
- You can use the search field to search for specific data model of interest.
- Step 5** Click the specific data model of interest to view more details.
- The data model is displayed in a hierarchical tree structure making it easier to navigate and understand the relationships between different YANG modules, containers, leaves and leaf lists. You can apply filters to further narrow down the data model definitions for the selected platform and release based on status such as deprecated, obsolete and unsupported nodes.
- You can also click the **Download** icon to export the data model information in Excel format.
- This visual tree form helps you get insights into the nodes that you can use to automate your network.
- The data models on Cisco Feature Navigator is regularly updated based on IOS XR release. If you encounter any problem or have suggestions for improvements, share your experience using [Send us your feedback](#) link.
-

Access data models from GitHub

To access the data models from GitHub repository, you can use these steps:

Procedure

- Step 1** Go to the [GitHub](#) repository for data models.
- On the repository page, you will find a list of folders based on IOS XR releases.
- Step 2** Navigate to the release folder of interest to view the list of supported data models and their definitions. For example, if you want to access the data models for IOS XR release 7.10.1, click on the folder named 7.10.1.
- Inside the folder, you will find a list of YANG files representing different data models.
- Step 3** Click on the YANG file you want to access to view its contents.
- You can also click on the **Raw** button to see the raw code or use the **Download** button to download the file to your computer.
- Each data model defines a complete and cohesive model, or augments an existing data model with additional XPath. To view a comprehensive list of the data models supported in a release, navigate to the **Available-Content.md** file in the repository. The unsupported sensor paths are documented as deviations. For example, `openconfig-acl.yang` provides details about the supported sensor paths, whereas `cisco-xr-openconfig-acl-deviations.yang` shows the unsupported sensor paths for `openconfig-acl.yang` model.
- Step 4** Repeat the above steps for other versions or data models of interest.

The GitHub repository for IOS XR data models is regularly updated based on release. You can also contribute to the repository by submitting pull requests, opening issues if you encounter any problems or have suggestions for improvements.

Get started with IOS XR YANG data models

Here is a generic outline of the steps involved in programmatically configuring your router using YANG data models:

1. Enable network management protocol—Manage the router remotely using the protocols such as NETCONF or gRPC.
2. Install the necessary libraries and tools—Depending on the programming language you are using, you may need to install libraries or tools to programatically interact with the router. For example, if you are using Python, you might need to install the `ncclient` library.
3. Establish a session with the router—Use the programming language of your choice to establish a connection to the router using NETCONF or gRPC protocols. This involves providing connection parameters such as device IP address, username, password, and port number.
4. Retrieve the router capabilities—View the supported features and functionalities available on the router.
5. Create or modify configurations—Use YANG data models to create or modify the configuration on the router.
6. Apply the configuration—Push the updated configuration via the NETCONF or gRPC protocol to modify the router's running configuration to reflect the desired changes.
7. Validate the configuration—Verify that the changes are successfully applied. You can retrieve the running configuration or specific configuration parameters to ensure that the device is configured as intended.