



The bridge to possible

Cisco Public

Cisco YANG Suite

YANG API Testing and Validation Tool for IOS XE + more

Technical Decision Maker

Jeremy Cohoe, jcohoe@cisco.com

Enterprise Networking, Technical Marketing Engineering

October 2024

Empower your NetOps with Cisco YANG Suite

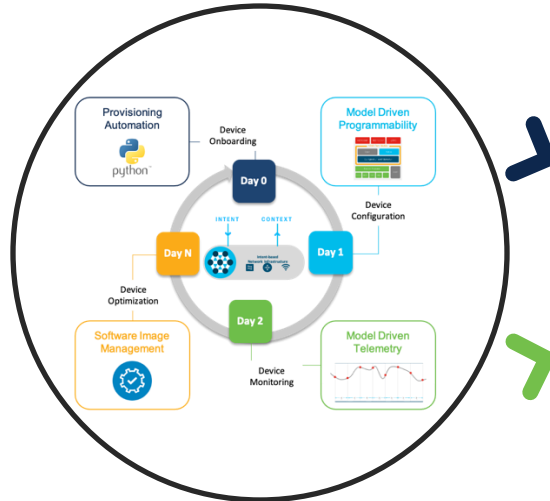
Best in class tooling for open programmability

Complex Network Operations

Cisco YANG Suite

Customer Benefits

Use Cases



**YANG
SUITE**
Cisco
YANG Suite



Simplified migration from CLI and SNMP



Toolkit to integrate with existing workflows



Guided workflow for all things YANG



API Testing and Validation Environment

Legacy Protocol Migration

Device Automation

Network Monitoring

Compliance and Coverage

<https://developer.cisco.com/yangsuite>

Simplify your device programmability journey with Cisco YANG Suite

Cisco YANG Suite



YANG API Testing and Validation Environment

Construct and test YANG based APIs over NETCONF,
RESTCONF, gRPC and gNMI

IOS XE / IOS XR / NX OS platforms

Top Screenshot: Explore YANG Models

YANG Suite / Exploring YANG / YANG set "C9300" / Modules

Select a YANG set: C9300 | Select YANG module(s): Cisco-IOS-XE-interfaces-oper | Load module(s)

Icon legend | Search XPath | Search nodes | Expand all nodes

Display schema nodes only | Display all nodes

Cisco-IOS-XE-interfaces-oper

- interfaces
 - interface
 - name
 - interface-type
 - admin-status
 - oper-status
 - last-change
 - if-index

Node Properties	
Name	statistics
NodeType	container
Description	A collection of interface-related statistics objects
Module	Cisco-IOS-XE-interfaces-oper
Revision	2020-07-01
Xpath	/interfaces/interface/statistics
Prefix	/interfaces-ios-xe-oper
Namespace	http://cisco.com/ns/yang/Cisco-IOS-XE-interfaces-oper

Bottom Screenshot: NETCONF

YANG Suite / NETCONF / YANG set "C9300" / Modules

YANG Set: C9300 | Module(s): Cisco-IOS-XE-interfaces-oper | Load Module(s)

NETCONF Operation: get | Device: JCOHOE-DMZ-C9300 | Edit Device | Open Device Window

Build RPC | Replays | RPC Options... | Run RPC(s) | Clear RPC(s)

Nodes

- Cisco-IOS-XE-interfaces-oper
 - interfaces
 - interface
 - name
 - interface-type
 - admin-status
 - oper-status
 - last-change
 - if-index
 - phys-address
 - higher-layer-if
 - lower-layer-if
 - speed
 - statistics

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get>
    <filter>
      <interfaces xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-interfaces-oper">
        <interface>
          <name/>
          <statistics/>
        </interface>
      </interfaces>
    </filter>
  </get>
</rpc>
```

Now Available !

developer.cisco.com/yangsuite

github.com/CiscoDevNet/yangsuite

Agenda

1

Introduction to Cisco YANG Suite

What's new: YANG Suite Innovations

2

IOS XE YANG API

Catalyst 9000 and IOS XE
YANG API with Telemetry

3

YANG Suite Overview

What's included

4

Getting Started

Docker and Python

5

Capabilities and Examples

Device & YANG File Management, Explore YANG
NETCONF, RESTCONF, gNMI
gRPC Telemetry, SNMP to YANG

6

Resources

YANG Suite, IOS XE API



Internal Programmability Teams Rooms

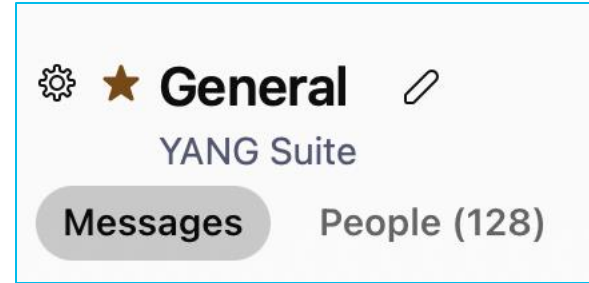
- **Support for Cisco Sales and Engineering:**
 - See internal resources on “SalesConnect” platform

- **Support for Cisco internal developers:**
 - See internal resources for “MTPE”

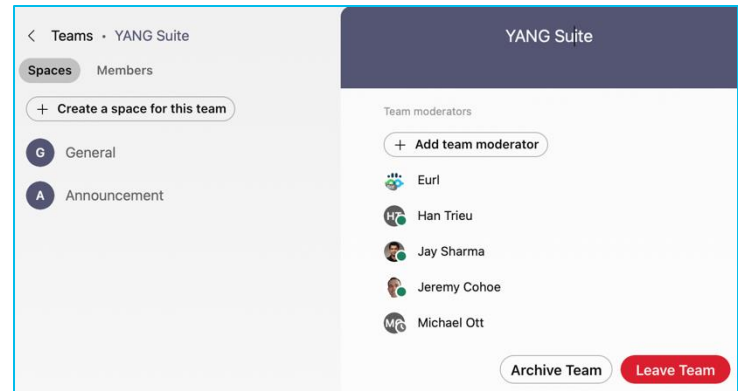


External Webex Teams Rooms

Anyone can join !



<https://eurl.io/#MaW78CeLS>



YANG Suite Innovations

YANG Suite project founder Michael retires

- Apoorva, Nhat, Han and team continue to maintain the YANG tooling and release integrations:

- Github
- YANG Catalog
- CFN
- OID Mappings

- YANG Suite is available for use on-demand from:
 - dCloud
 - DevNet Reservable Sandbox
- Available for install from Github/Docker or PyPi+python3



YANG Suite Update Schedule & Pre-releases

- When any bug fixes go into the internal YANG Suite, they are now automatically also pushed to the public external repository as well – in the “pre-release”.
- The “pre-release” code is promoted to mainline on the 28th of the month.
- We have created a feature request to enable installation from “pre” in the GUI however the current option is CLI only

Pre Release Versions [↗](#)

Manually specify pre-release version when using pip like in the example below to access the pre release and bug fix versions of yang suite and plugins:

```
docker container exec -it <yangsuite container ID> bash
pip install --upgrade --pre yangsuite-devices
exit

pip
source <your yangsuite venv>/bin/activate
pip install --upgrade --pre yangsuite-devices
```

Update all Plugins to Latest Release [↗](#)

```
pip3 install --upgrade yangsuite yangsuite-devices yangsuite-filemanager yangsuite-yangtree yangsuite-coverage ya ↗
```

<https://github.com/CiscoDevNet/yangsuite/blob/main/README.md#pre-release-versions>

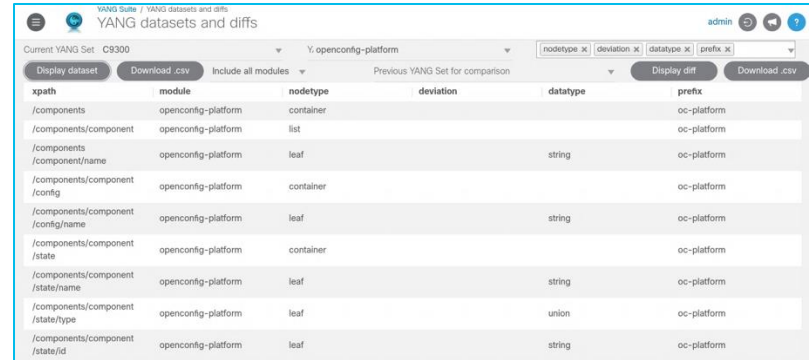
YANG datasets and diffs enhancements

Compare YANG, run diffs, display and export metadata to view see changes for various software releases

The YANG datasets and diff's plugin has been upgraded to enable comparison between complete YANG module set's and repositories. Previously this was enabled for a single YANG module only – Now it works with the entire set.

For example, a YANG set from release 17.6 can be compared against 17.9 to understand changes across all modules in the set.

All Native modules can also be compared, for example.



The screenshot shows a web interface for comparing YANG datasets. The current YANG set is 'CS300' and the previous set for comparison is 'Y. openconfig-platform'. The interface includes buttons for 'Display dataset', 'Download .csv', and 'Include all modules'. A table displays the following metadata:

xpath	module	nodetype	deviation	datatype	prefix
/components	openconfig-platform	container			oc-platform
/components/component	openconfig-platform	list			oc-platform
/components/component/name	openconfig-platform	leaf		string	oc-platform
/components/component/config	openconfig-platform	container			oc-platform
/components/component/config/name	openconfig-platform	leaf		string	oc-platform
/components/component/state	openconfig-platform	container			oc-platform
/components/component/state/name	openconfig-platform	leaf		string	oc-platform
/components/component/state/type	openconfig-platform	leaf		union	oc-platform
/components/component/state/id	openconfig-platform	leaf		string	oc-platform

YANG Suite +(mac) pip install + HTTPS support

YANG Suite can now be accessed using HTTPS with TLS certificates when installed using the pip package manager mechanism.

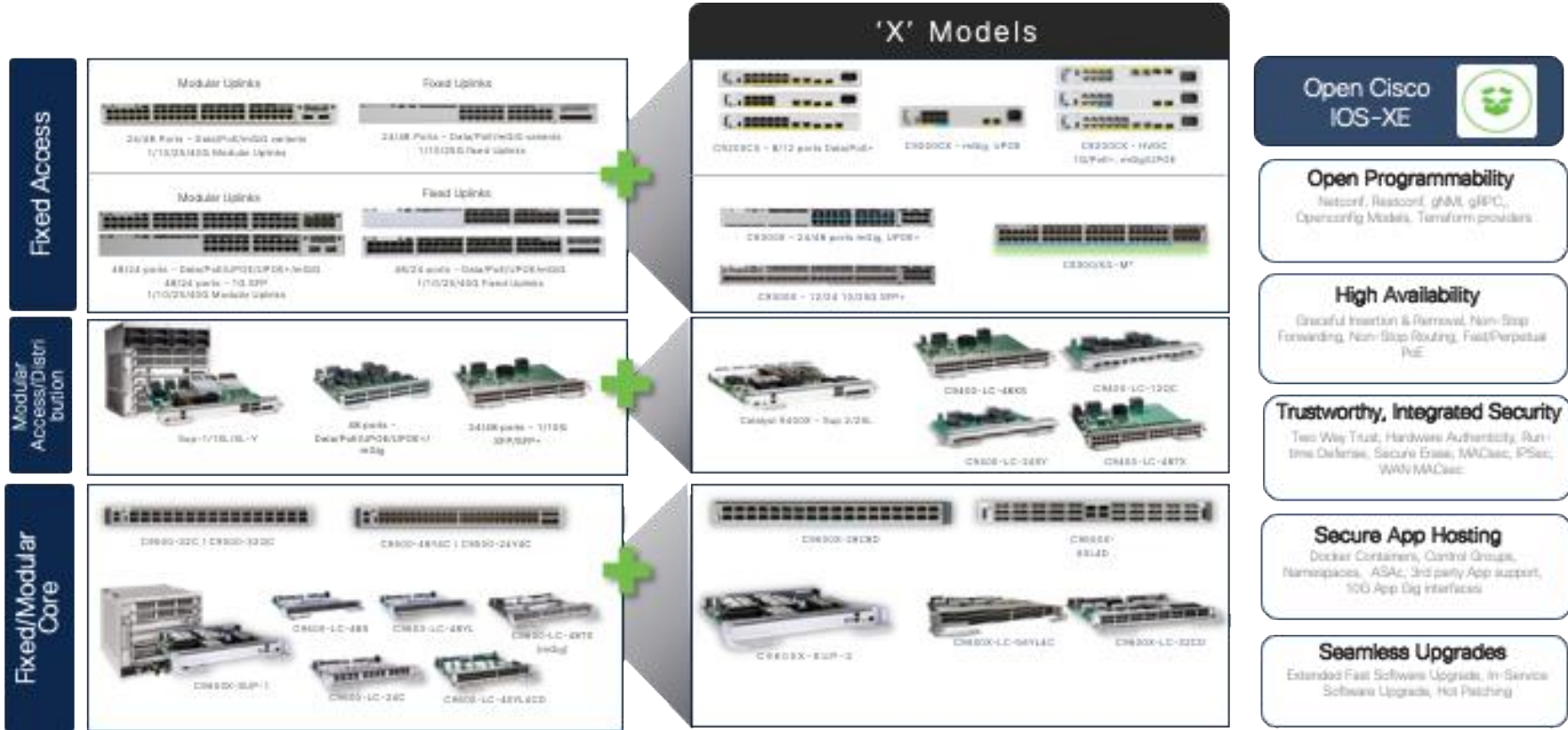
The 3-Docker container solution can still be used when HTTPS is required and when the docker framework is the preferred solution

You can now provide the flag “—https” option when starting YANG Suite along with the required TLS certificates to enable the secure connection into the tool

IOS XE YANG API Overview

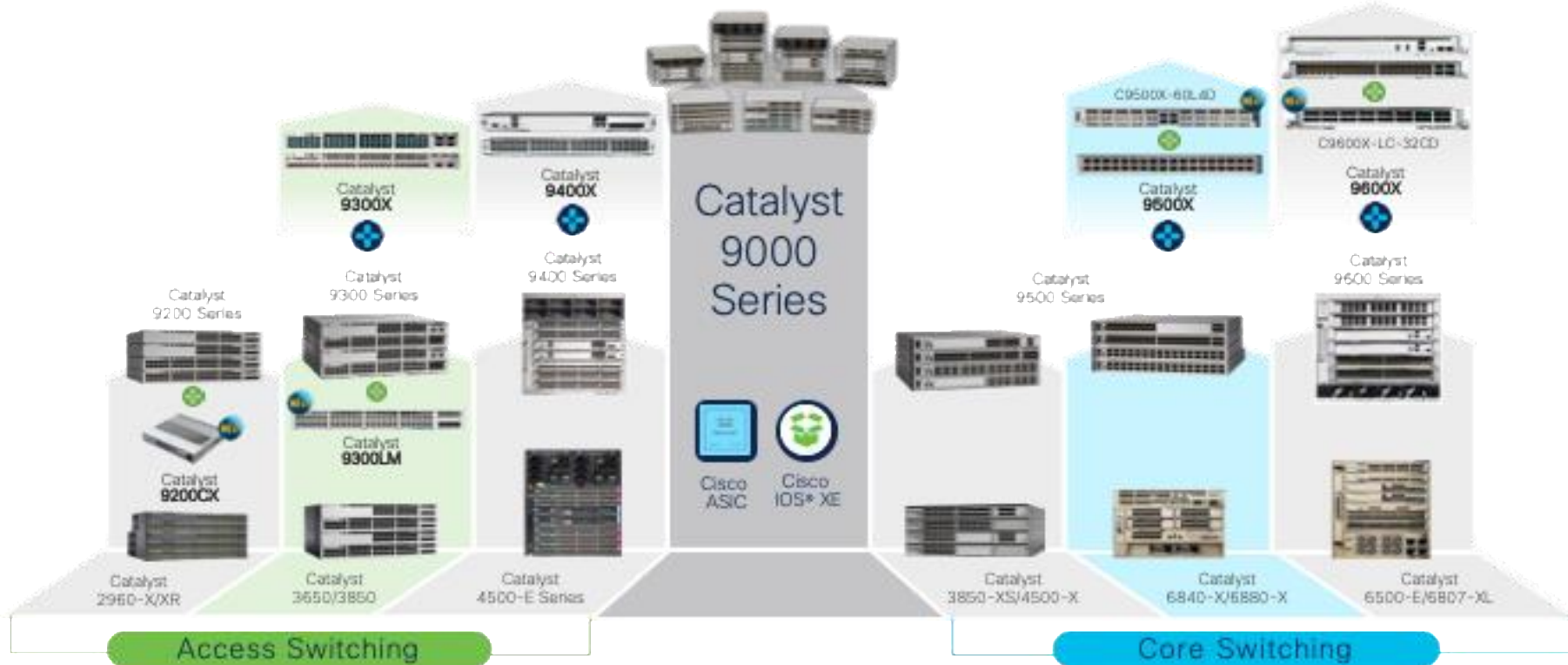
Cisco Catalyst 9000 Switching

Comprehensive campus switching portfolio for evolving needs of modern businesses

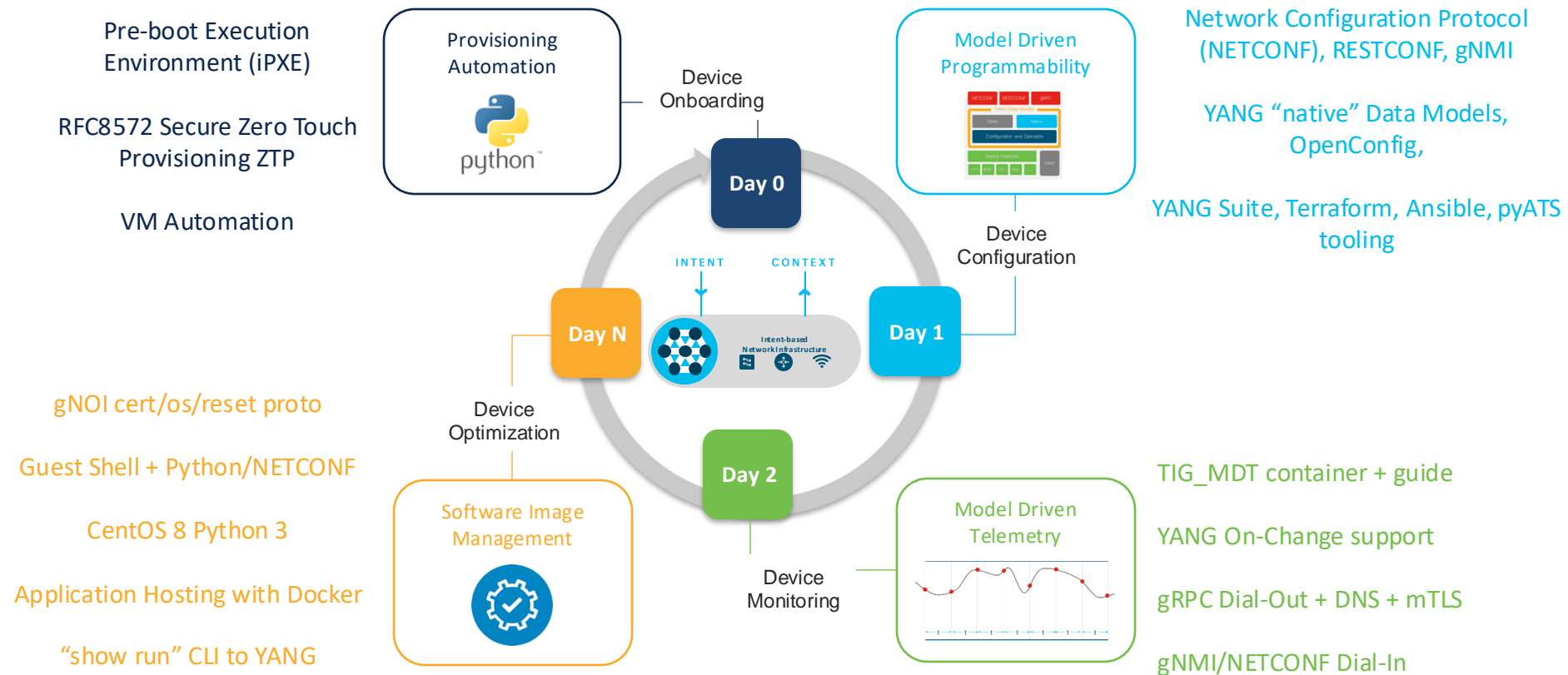


Cisco Catalyst 9000 Switching Portfolio

One Family from Access to Core



IOS XE Programmability & Automation Lifecycle



Cisco IOS XE Programmability and Telemetry “Stack”

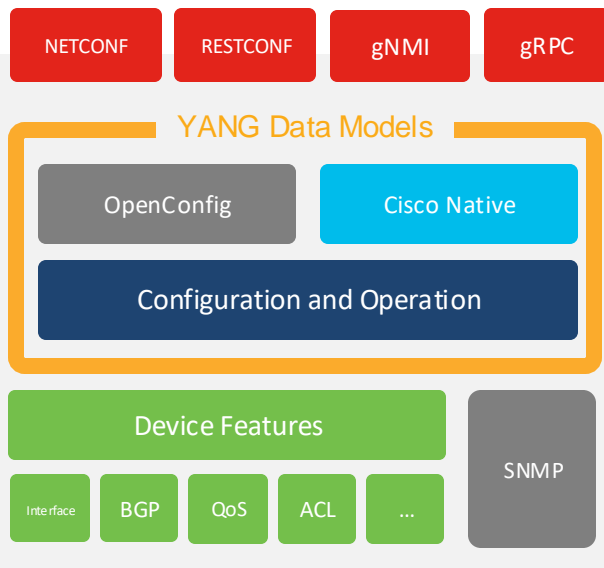
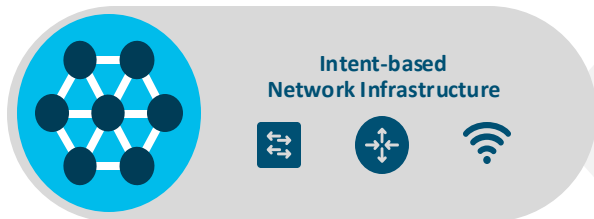
CLI

SNMP

WebUI

The NETCONF, RESTCONF, gNMI and gRPC are programmatic interfaces that provide **additional** methods for interfacing with the Cisco IOS XE device – Just like the CLI, SNMP, and WebUI is used for configuration changes and operational metrics so can the programmatic interfaces of NETCONF, RESTCONF, gNMI, and gRPC.

YANG data models define the data that is available for configuration and streaming telemetry

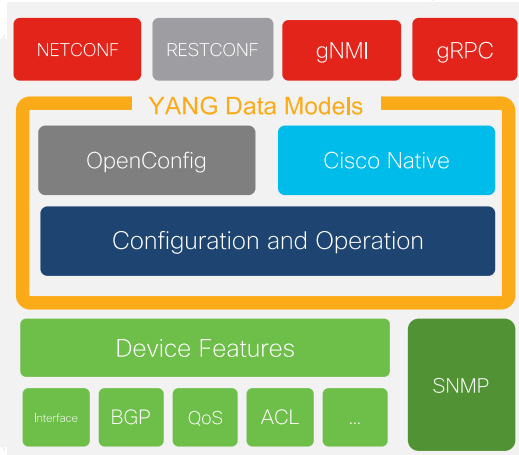


Model Driven Telemetry Interfaces

↔ Dial In: Collector establishes a connection to the device then subscribes to telemetry (pub/sub)

← Dial Out: Telemetry is pushed from the device to the collector based off configuration (push)

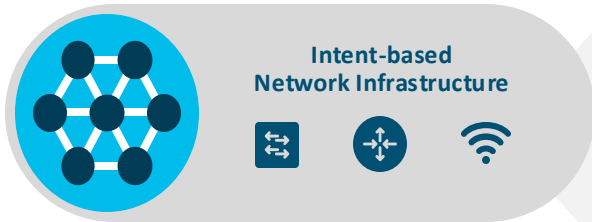
Publication / Subscription



XML, JSON, proto and kvGPB encoding

Consistent YANG data models between interfaces

On-change event and time-based publication options

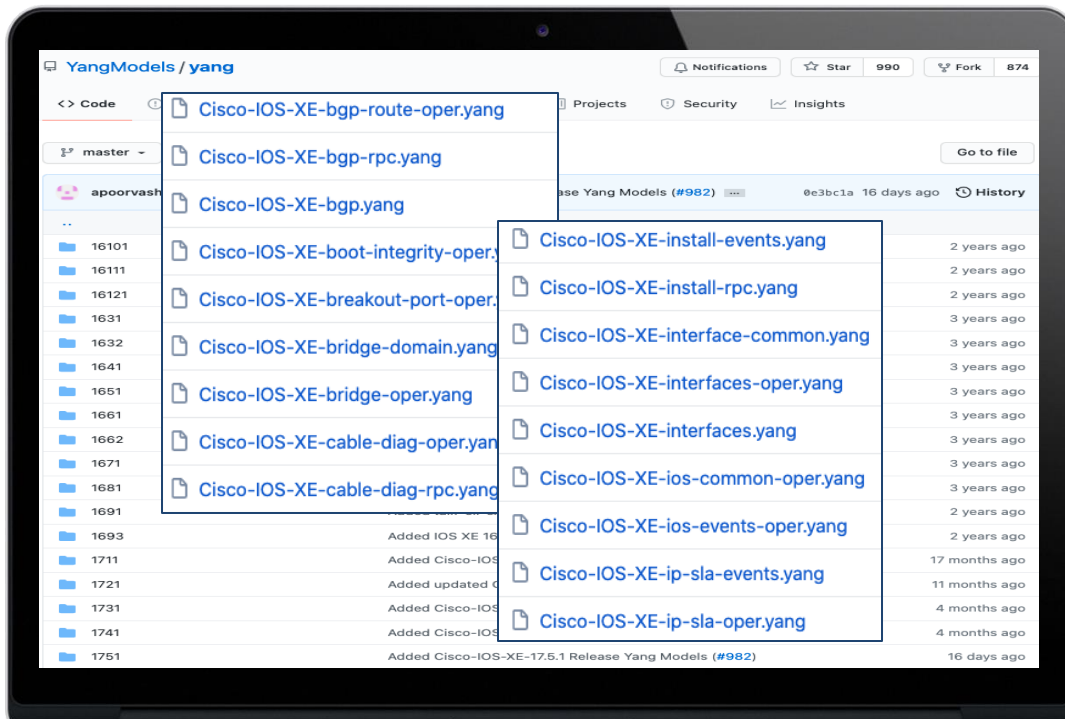


Cisco IOS XE - YANG model API documentation



- There are 10 types of data models including config, oper, actions, and deviations
- RFC 7950 YANG data modelling language are the API definitions for IOS XE
- The YANG modules are available for download from the API and are also published on Github.com
- Notable modules are listed below for the running-config, feature oper, actions and event notifications

TYPE	YANG module name.yang	Description
1	Native configuration model	Cisco-IOS-XE-native running-config
2	Operation	Cisco-IOS-XE-(feature)-oper Feature operational data
3	Configuration	Cisco-IOS-XE-(feature)-cfg Feature configuration, independent of "native" model
4	Events	Cisco-IOS-XE-(feature)-events Telemetry Events that can be triggered
5	RPC	Cisco-IOS-XE-(feature)-rpc Actions that can be performed
6	Deviation	Cisco-IOS-XE-(feature)-deviation Device implementation deviation from module
7	Types	Cisco-IOS-XE-(feature)-types Types - Imported by other modules
8	Obsolete	Cisco-IOS-XE-(feature)-obsolete Obsolete should not be implemented
9	Common	Cisco-IOS-XE-(feature)-common Common - Imported by other modules
10	Abstractions	OpenConfig-(feat, 'a) & Cisco-evpn-service abstraction for EVPN, OpenConfig config & oper



The YANG models are available for download directly from the running IOS XE device's NETCONF, RESTCONF, or gNMI API, and from:

<https://github.com/YangModels/yang/tree/main/vendor/cisco/x>

Getting started with NETCONF on IOS XE

Enable NETCONF-YANG and set the required AAA configuration

```
netconf-yang
aaa new-model
aaa authentication login default local
aaa authorization exec default local
username netconf privilege 15 password 0 netconf
```

Once enabled, validate by connecting to the NETCONF interface with SSH:

```
ssh -p 830 netconf@<IOS XE IP address> -s netconf
```

```
jcohoe@JCOHOE-M-C6NA ~ % ssh -p 830 admin@jcohoe-c9300 -p netconf
admin@jcohoe-c9300's password:
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
<capability>urn:ietf:params:netconf:capability:writable-running:1.0</capability>
<capability>urn:ietf:params:netconf:capability:rollback-on-error:1.0</capability>
```

Refer to Programmability Configuration Guide at https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/173/b_173_programmability_configuring_yang_datamodel.html#id_84436

Enable NETCONF examples

IOS XE:

```
configure terminal > netconf-yang
aaa new-model
aaa authentication login default local
aaa authorization exec default local
username netconf privilege 15 password 0 Netconf
```

IOS XR:

```
configure terminal > ssh server v2
ssh server netconf
netconf agent tty
netconf-yang agent ssh
```

NX-OS:

```
configure terminal > feature netconf
```

Once enabled, validate by connecting to the NETCONF interface using SSH:

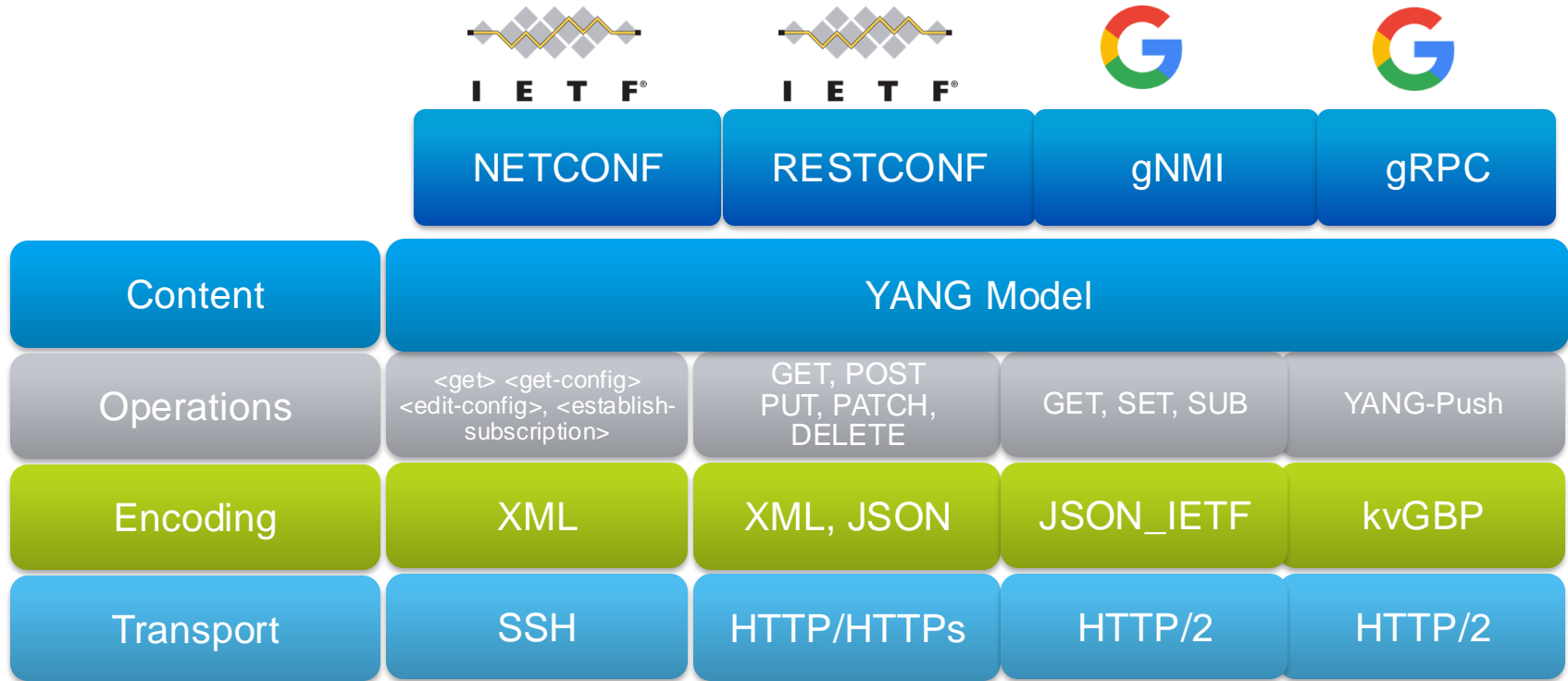
```
ssh -p 830 -s netconf user@ip
```

Refer to Programmability Configuration Guides for more details

API Operations

NETCONF	RESTCONF	gNMI	gRPC
<get-config>, <get>	GET	GET	
<edit-config> (operation="create")	POST	SET = Replace	
<edit-config> (operation="replace")	POST, PATCH	SET = Update	
<edit-config> (operation="delete")	DELETE	SET = Delete	
<establish-subscription> On-Change + Periodic		SUBSCRIBE On-Change + Periodic	YANG push On-Change + Periodic
Other RPC's			

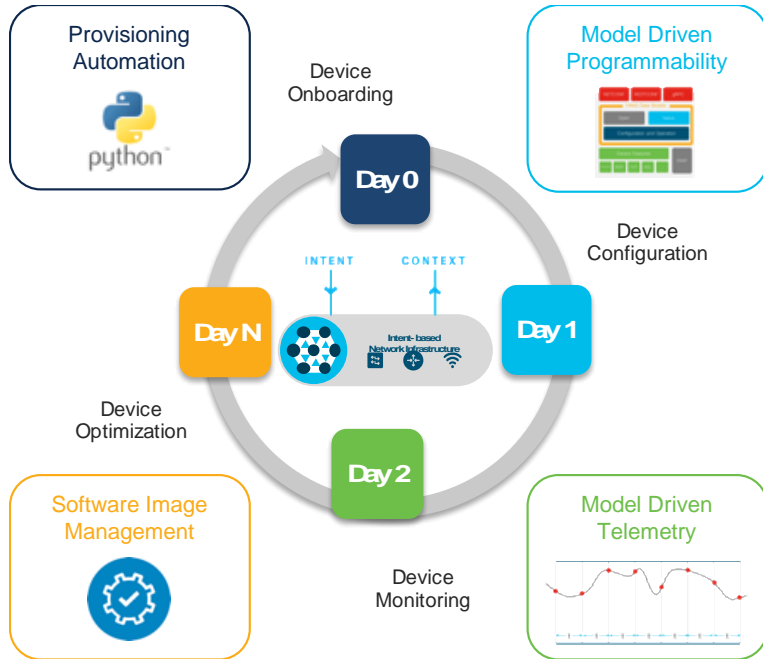
API Interfaces – Including Telemetry



YANG Suite Overview

Cisco IOS XE Automation

Automated Lifecycle powered by YANG Suite



- Day-0 Installation
- Day-1 Configuration and Operations
- Day-2 Optimization and Compliance
- Day-N Patching and Maintenance

Use Cases

1

Legacy Migration

As we see increased adoption of Catalyst 9000 portfolio from legacy devices, YANG Suite helps facilitate migration from CLI to YANG

2

Device Automation

With new versions of IOS XE, XR, and NX-OS Software, YANG Suite allows exploring the models and provide toolchain to allow automation of such features

3

Network Telemetry

To leverage the additional benefits of Model Driven telemetry, YANG Suite provides a path to migrate from SNMP towards YANG based Model Driven Telemetry

4

Compliance and Coverage

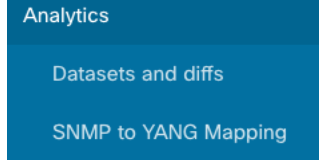
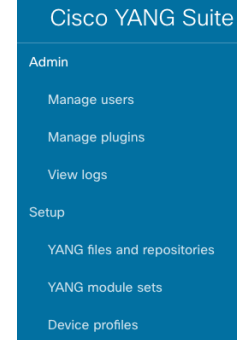
YANG Suite helps understand model coverage for existing features and helps NetOps in network validation

What's Included

- Initial Release:
 - Plugin and YANG File Manager, Datasets and diffs
 - Device Manager
 - NETCONF (Python), gRPC Telemetry
 - Docker install support with HTTPS
- Second Release:
 - RESTCONF
 - gNMI
 - Python Integrations
- Third Release:
 - gRPC Telemetry with TLS Support
 - SNMP OID to YANG Xpath Mapping
 - Ansible Integrations
 - Pip install support

Core plugins

Additional plugins



Plugins

Plugin	Description	Release
yangsuite	Core application, plugin management and API libraries	Core
yangsuite-devices	Add and edit Cisco IOS XE, XR and NX devices	Core
yangsuite-netconf	NETCONF client including telemetry	Core
yangsuite-filemanager	Download and manage YANG modules	Core
yangsuite-yangtree	Explore YANG modules in detail	Core
yangsuite-coverage	Analytics > Datasets & Diffs	Core
yangsuite-grpc-telemetry	gRPC Telemetry receiver	V1
yangsuite-restconf	RESTCONF client	V2
yangsuite-gnmi	gNMI client including telemetry	V2
yangsuite-coverage	SNMP to YANG	V3

Cisco EULA License

<https://github.com/CiscoDevNet/yangsuite/blob/main/LICENSE>

Cisco YANG Suite User Agreement

YANG Suite users must agree to the "Cisco End User License Agreement" and "Privacy Statement".

Choose to accept or decline "Cisco End User License Agreement".


Decline Accept

Choose to accept or decline "Cisco Online Privacy Statement".

Decline Accept

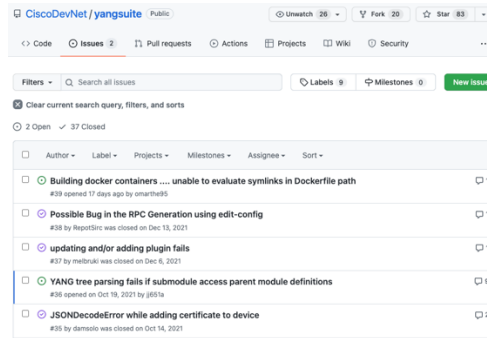
Submit

```
root@02a59b6c5b43:/# cat /root/yangsuiteenv/lib/python3.8/site-packages/yangsuite/yangsuite.ini
[DEFAULT]
data_path = /root/yangsuiteenv/ys-data
port = 8480
secret_key = !v*%s%xb_qrnmf&%e14n)m*4$i44n(cj1ny1=5+epjhya1^wf
static_root = /root/.local/share/yangsuite/static
allowed_hosts = *
certificate_path = /path/to/certificate/file.pem
private_key_path = /path/to/private_key/file.pem
settings_module = yangsuite.settings
eula_agreement = accepted
dna_advantage = detect
```

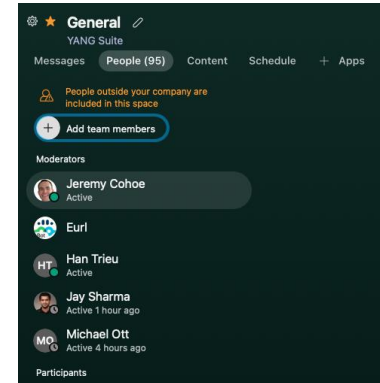


YANG Suite Usage

This free tool has been available for over a year and helps customers and partners to increase adoption and less friction with validation of API's and telemetry



<https://eurl.io/#MaW78CeIs>



~180 clones in 14-day period

Active issues and discussion forums

What's Included – YANG Suite Core

Plugin Manager: Additional plugins are installed and updated through the plugin manager from the online Python Package Index (PyPi)

YANG File Manager: The YANG Sets and Repositories allows for downloading and sorting of the YANG models

Device Manager: Support for adding multiple devices and specifying which of the programmatic interfaces are enabled

Package name	Description	Installed version	Latest version
yangsuite	Core application logic for YANG Suite. Capable of dynamic discovery of installed application plugins. Provides common library APIs for logging, filesystem access, GUI appearance and behavior, and client-server communication.	2.7.9	2.7.9
yangsuite-devices	Provides common infrastructure for definition and management of network device profiles. Manages device profile validation in the form of connectivity and credential checks.	2.8.3	2.8.3
yangsuite-filemanager	Provides quick, low-overhead parsing of YANG (RFC 6020, RFC 7950) models and identification of their interdependencies. Manages YANG file repositories and sets of YANG files within these repositories. Provides UI and APIs for file upload to YANG Suite.	1.8.1	1.8.1
yangsuite-yangtree	Manages loading, caching, and validation of YANG (RFC 6020, RFC 7950) models. Represents parsed YANG models as Python dicts and JavaScript trees. Adds GUI for traversing, searching, and inspecting YANG model trees.	1.19.1	1.19.1

Package name	Description	Installed version	Latest version
yangsuite-coverage	Checks YANG model coverage based on Cisco CLI config	2.2.8	2.2.9
yangsuite-grpc-telemetry	gRPC Telemetry support for YANG Suite	0.7.1	0.7.1
yangsuite-netconf	Adds NETCONF protocol (RFC 6241, RFC 7950) support to YANG Suite. Allows the user to build NETCONF RPC messages and execute them on live network devices that support NETCONF. Users can also subscribe to NETCONF event notifications (RFC 5277) from devices with this capability.	1.15.3	1.15.3

YANG Suite / Device profiles
Manage device profiles

Select a device profile

sandbox-isoxe-latest-1.cisco.com

New Device Profile

Fields marked with * are required.

General Info

Profile Name *

Description

Address *

Username

Password

Timeout * 30

Buttons: Create new device, Check selected device's reachability, Clone selected device, Edit selected device, Delete selected device, Create default Repository and Yangset

What's Included – YANG Suite Additional Plugins

Datasets and Diffs (Compare): Load various YANG models to display and compare or export the metadata **NETCONF:** Build and generate XML payload and interact with the device using NETCONF protocol

gRPC Telemetry: Listen for Model Driven Telemetry (streaming telemetry) on the specified TCP port

The screenshot displays the YANG Suite interface with three main sections:

- Datasets and Diffs (Compare):** A table showing YANG models and their metadata. The current YANG Set is 'Cisco-IOS-XE-process-cpu-oper'. The table lists various xpaths and their corresponding modules, datatypes, prefixes, keys, and operations.
- NETCONF:** A section for interacting with a device. It shows the current YANG Set as 'Cisco-IOS-XE-interfaces-oper' and the device as 'sandbox-iosxe-latest-1.cisco.com'. It includes buttons for 'Load Module(s)', 'Run RPC(s)', and 'Clear RPC(s)'. A tree view shows the YANG model structure, and a text area displays the generated XML payload for a 'get' operation.
- gRPC Telemetry:** A section for listening for streaming telemetry. It shows the current configuration: 'Listen at IP address... 127.0.0.1' and 'Listen at port 57344'. It includes a 'Stop telemetry receiver' button and a 'Clear output' button. The output shows the received telemetry data for the 'Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization' path.

xpath	module	datatype	prefix	key	operations
/cpu-usage	Cisco-IOS-XE-process-cpu-oper		process-cpu-ios-xe-oper		get
/cpu-usage/cpu-utilization	Cisco-IOS-XE-process-cpu-oper		process-cpu-ios-xe-oper		get
/cpu-usage/cpu-utilization/five-seconds	Cisco-IOS-XE-process-cpu-oper	uint8	process-cpu-ios-		
/cpu-usage/cpu-utilization/five-seconds-intr	Cisco-IOS-XE-process-cpu-oper	uint8	process-cpu-ios-		
/cpu-usage/cpu-utilization/one-minute	Cisco-IOS-XE-process-cpu-oper	uint8	process-cpu-ios-		

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get>
    <filter>
      <interfaces xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-interfaces-oper"/>
    </filter>
  </get>
</rpc>
```

```
Node : C9300
Subscription : 202
Path : Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization
/five-seconds : 0
```

What's Included – YANG Suite Additional Plugins

RESTCONF: OpenAPI “Swagger” user interface based on YANG set

gNMI: A complete gNMI client supporting GET, SET, and Subscribe operations, and encoding options

OpenAPI v3.0.3 3.0.3 OAS3

HOST DESTINATION: https://10.1.1.5:443 (proxy through YANG Suite server)

Servers

/restconf/proxy/https://10.1.1.5:443/restconf - YANG SUITE Proxy RESTCONF API

default

GET	/data/Cisco-IOS-XE-interfaces-oper:interfaces
GET	/data/Cisco-IOS-XE-interfaces-oper:interfaces/interface
GET	/data/Cisco-IOS-XE-interfaces-oper:interfaces/interface={interface-name}
GET	/data/Cisco-IOS-XE-interfaces-oper:interfaces/interface={interface-name}/interface-type
GET	/data/Cisco-IOS-XE-interfaces-oper:interfaces/interface={interface-name}/admin-status

YANG Suite / gNMI / YANG set ** / Modules

gRPC Network Management Interface (gNMI)

admin

YANG Set: c9300-default-yangset Module(s): Cisco-IOS-XE-interfaces-oper

Device: C9300

gNMI Operation: Get Set Subscribe All Config State Operational

Origin: Openconfig RFC 7951 Other

Encoding type: JSON_IETF JSON Prefixing

Search XPath... Show Legend Build JSON Clear Values Run RPC(s) Clear RPC(s)

Nodes

- Cisco-IOS-XE-interfaces-oper
 - interfaces
 - interface
 - name
 - interface-type
 - admin-status
 - oper-status
 - last-change
 - if-index

```
path {
  origin: "rfc7951"
  elem {
    name: "Interfaces"
  }
  elem {
    name: "Interface"
    key {
      key: "name"
      value: ""
    }
  }
}
encoding: JSON_IETF
```

Getting Started Workflow

Docker vs Python install

There are two supported YANG Suite installations using either a Docker container or by running the python application natively within an existing operating system

Docker: pre-configured 3-container supporting HTTPS with reverse proxy

Python: native python application that is deployed on existing OS or VM



<https://www.docker.com>

<https://www.python.org>

YANG Suite Docker Containers

Docker: System Requirements

YANG Suite runs within Docker and is installed using the python pip package system

Mac/Linux/Windows

- Docker Engine
- Docker Compose
- Git
- OpenSSL
- Browser: Chrome/Firefox/Safari

4GB Memory for Docker
Port 80 and 443 by default



miott Cisco Employee

02-17-2021 11:30 AM

YANG Suite docker install on Windows 10

The YANG Suite team will not be distributing a docker image to dockerhub but there is a docker recipe to build your own docker containers using docker-compose and dockerfile which can be retrieved from the [CiscoDevNet/yangsuite](https://github.com/CiscoDevNet/yangsuite) Git repository. Building this container on Windows 10 has special challenges. Here are a couple things Windows 10 users had to do to get this working.

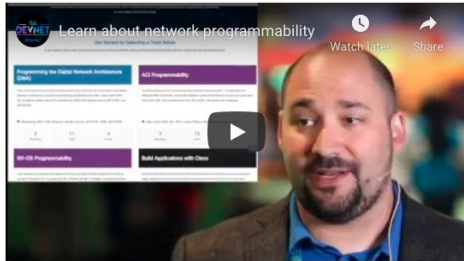
Doing the "git clone" has an issue with CRLF and you can workaround it with the following command in the gitshell:

```
git config --global core.eol lf
git config --global core.autocrlf input
find . -type f -print0 | xargs -0 dos2uni
```

Detailed installation instructions are on the [Community Forum](#)

Getting Started with Docker and Git on DevNet

Learn at your own pace with free Learning Labs



Cisco DevNet provides tutorial-style walkthroughs of our current programmable technology, including sandboxes with admin access to real Cisco hardware. Find out why the DevNet community comes back to these Labs, over and over again.

[Learn how to setup a local developer environment.](#)

1 Docker 101

2 Get or create a Docker image

3 Build a Docker image

4 Build a web server container

5 How to remove existing Docker Container and Images

1 Introduction

2 Step 1: Intro to Version Control Systems

3 Step 2: Basic Git workflows

4 Step 3: The DevNet sample code workflow

5 Where to go to learn more

Docker 101 <https://developer.cisco.com/learning/lab/docker-101/step/1>

Git <https://developer.cisco.com/learning/lab/git-basic-workflows/step/1>

FAQ – Internal version only, being added to <https://developer.cisco.com/yangsuite>

Docker Containers

The main “yangsuite” docker container has the application and runtime data

This is an Ubuntu container

The YANG files and configuration is backed up with the yangsuite-backup container

The nginx container is the HTTPS front-end to YANG Suite

This is an Alpine Linux container

```
auto@pod2-xelab:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS          PORTS
011e80b4ea18   nginx:latest   "/docker-entrypoint...." 44 seconds ago Up Less than a second 0.0.0.0:8443->8443/tcp,
docker_nginx_1
1b9b91526a1a   backup:latest  "/code/run_cron.sh"     5 hours ago   Up Less than a second  docker_backup_1
18331dba06e5   yangsuite:latest  "/yangsuite/migrate_..." 5 hours ago   Up 1 second    0.0.0.0:443->443/tcp, :
```

- yangsuite
- yangsuite-backup
- nginx



NGINX



Quick Start with Docker

1. git clone <https://github.com/CiscoDevNet/yangsuite>
2. Build certificates: `cd yangsuite/docker/ ; ./gen_test_certs.sh`
3. `docker-compose up`

```
jcohoe@JCOHOE-M-C6NA docker % docker-compose up --build
Building yangsuite
Step 1/19 : FROM ubuntu:18.04
--> 2c047404e52d
Step 2/19 : ENV DOCKER_RUN true
--> Using cache
--> 3b8e0efd0cfd
Step 3/19 : ARG PY=python3.6
```

2. Access the tool at <https://localhost>

Cisco YANG Suite User Agreement

YANG Suite users must agree to the "Cisco End User License Agreement" and "Privacy Statement".

Choose to accept or decline "Cisco End User License Agreement".

Decline Accept

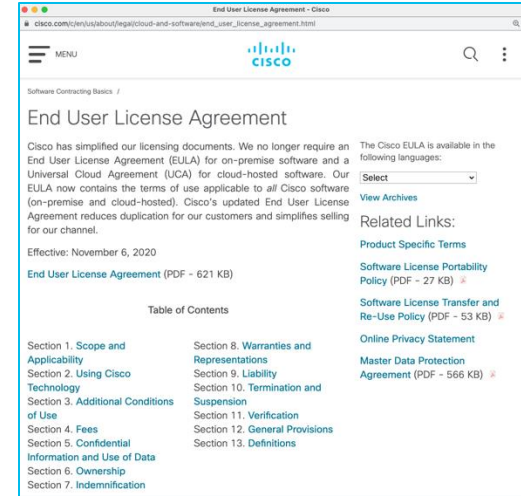
Choose to accept or decline "Cisco Online Privacy Statement".

Decline Accept

Submit

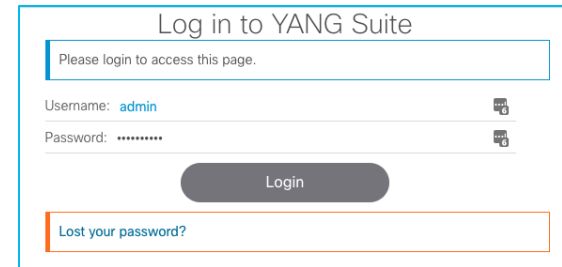
https://www.cisco.com/c/en/us/about/legal/cloud-and-software/end_user_license_agreement.html
<https://www.cisco.com/c/en/us/about/legal/privacy-full.html>

3. Review and accept license and privacy policy



4. Login to the tool at <http://localhost>

Default credentials in the Dockerfile
admin : superuser



YANG Suite v3 and Docker

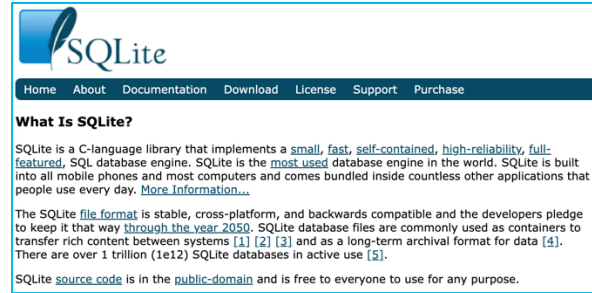
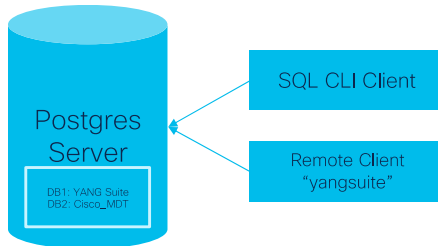
Release 3 brings enhancements to the Docker environment including:

1. Upgraded Operating System Ubuntu 18 to Ubuntu 20
2. Python 3.8 is now used
3. Option to use PostgreSQL instead of SQLite
4. Sendmail included for password reset notification emails

Option to use PostgreSQL instead of SQLite

Default option has been to use a local .sqlite file

Now a local or remote PostgreSQL database can be used



```
root@e106c4e550f9:~/ys-data#
root@e106c4e550f9:~/ys-data# pwd
/root/ys-data
root@e106c4e550f9:~/ys-data#
root@e106c4e550f9:~/ys-data# ls
db.sqlite3  devices  logs  users  yangsuite.canary
root@e106c4e550f9:~/ys-data#
root@e106c4e550f9:~/ys-data#
root@e106c4e550f9:~/ys-data#
```

Pip install support

Requirements:

- 64-bit Windows10, Mac, Ubuntu, CentOS, or FreeBSD
- 8 GB Memory, Python 3.7 & 3.8
- Prerequisite: pip3 in Linux and Windows

```
$ pip3.8 install yangsuite
$ yangsuite -i
```



Ensure pre-requisites are installed in a virtual environment as a non-root user

Ubuntu Linux example:

```
$ apt-get install git openssh-client iputils-ping sqlite3 snmp python3.8 python3-pip
```

Windows:

Install python3 and python3-pip from python.org

Recommended: Install Windows Subsystem for Linux (WSL).

Note: Python3 comes with WSL, but not python3-pip or Django>=3.2

<https://learn.microsoft.com/en-us/windows/wsl/about>

Mac:

Install python3

The Python Package Index (PyPI) is a repository of software for Python

<https://pypi.org/project/yangsuite/>

Pip Install Support and Details

- Requirements:
 - Install recommended or supported Python 3.8
 - Install YANG Suite as a non-root user
 - Run YANG Suite in a virtual environment as a non-root user
- Linux notes:
 - Ubuntu 20 includes 3.8 by default
 - Ubuntu 22 Guide (uses a higher version of Python by default, which needs to be downgraded)
 - <https://www.linuxcapable.com/how-to-install-python-3-8-on-ubuntu-22-04-lts/>

Note: Due to the way most Linux distributions are handling the Python 3 migration, Linux users using the system Python without creating a virtual environment first should replace the `python` command in this tutorial with `python3` and the `python -m pip` command with `python3 -m pip --user`. Do *not* run any of the commands in this tutorial with `sudo`: if you get a permissions error, come back to the section on creating virtual environments, set one up, and then continue with the tutorial as written.

<https://packaging.python.org/en/latest/tutorials/installing-packages/>

Virtual Environment:

```
$ mkdir yangsuite ; cd yangsuite
$ python3.8 -m venv .
$ source ./bin/activate
(yangsuite) $ pip3.8 install yangsuite
(yangsuite) $ yangsuite
                Answer the prompted questions !
                or
(yangsuite) $ yangsuite &
                Start in the background
(yangsuite) $ yangsuite -l
                to see YS settings
```

Python Virtualenv Installation

WARNING: HTTPS support is not available unless a web server is installed like NGINX and frontended with uwsgi. Instructions for that can be found searching online.

- Create a Python 3.6, 3.7, or 3.8 virtual environment and activate it.
- Type `pip install yangsuite[core]`
- Use `yangsuite` to start the yangsuite server
- User is prompted to set superuser credentials.
- Use `ctrl-c` to stop the yangsuite server
- Connect to <http://localhost:8480>.
- Login using user set at install.
- Install other plugins using Admin-->Manage plugins page.

YANG Suite on Mac with pip

Install home brew

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

```
$ brew install pyenv
```

```
$ pyenv install 3.8 (will get latest (currently 3.8.15))
```

or

```
$ pyenv install 3.8.12 (or latest)
```

```
$ ~/.pyenv/versions/3.8.12/bin/python -m venv yangsuiteenv
```

```
$ source yangsuiteenv/bin/activate
```

```
$ pip install pip wheel setuptools --upgrade
```

```
$ pip install yangsuite
```

```
$ yangsuite
```

fill in prompts:

The path to the data is absolute so `/Users/username_here/youngsuiteenv/` is recommended

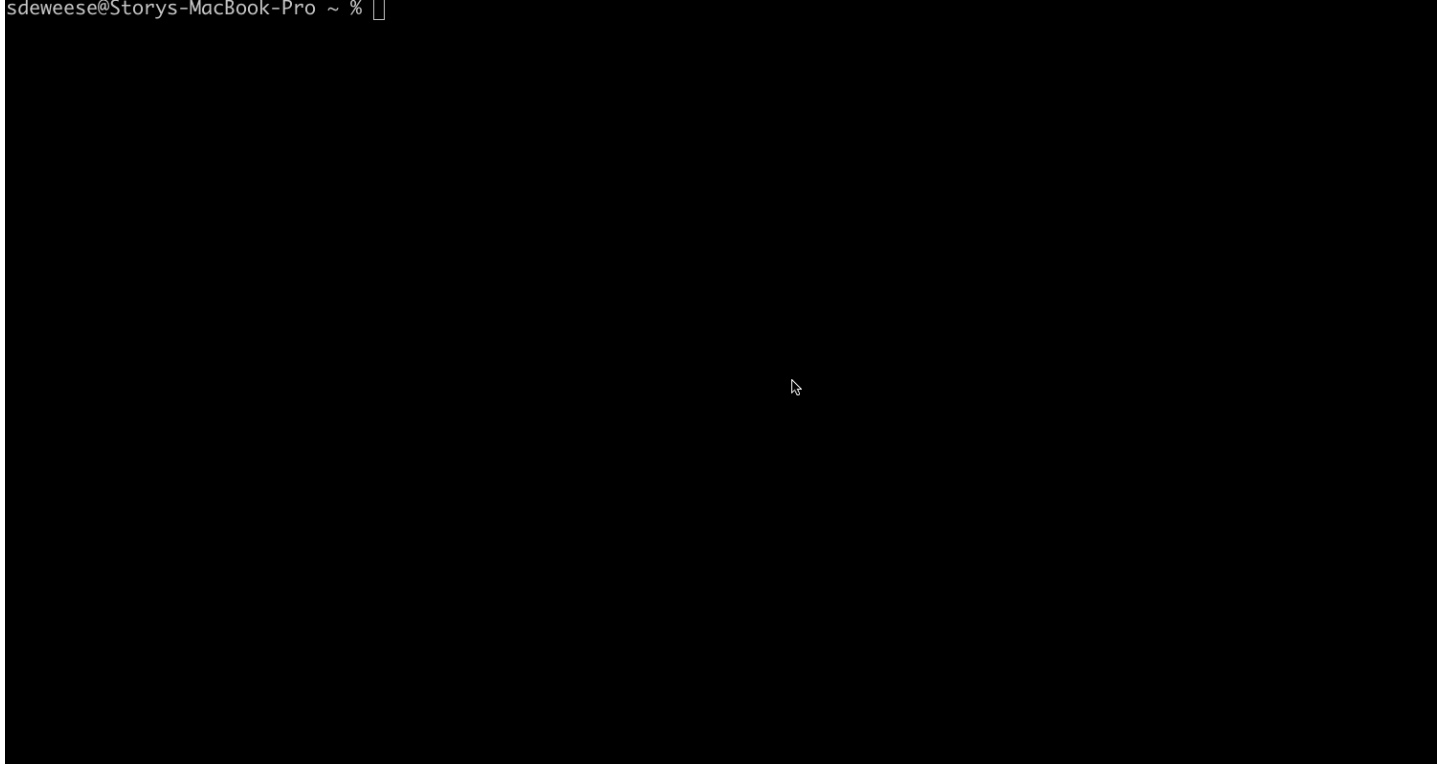
Next,

- Install additional plugins from GUI
- Add devices, add YANG modules, and start sending API calls!



Pip Install example on Mac

```
sdeweese@Storys-MacBook-Pro ~ %
```



YANG Suite install on Ubuntu 22

Install PyENV from https://kfields.me/blog/pyenv_on_ubuntu_22

```
sudo apt-get update; sudo apt-get install make build-essential libssl-dev  
zlib1g-dev \  
libbz2-dev libreadline-dev libsqlite3-dev wget curl llvm \  
libncursesw5-dev xz-utils tk-dev libxml2-dev libxmlsec1-dev libffi-dev  
liblzma-dev
```

```
curl https://pyenv.run | bash
```

```
pyenv install 3.8.12
```

```
~/pyenv/versions/3.8.12/bin/python -m venv yangsuiteenv  
source yangsuiteenv/bin/activate
```

```
pip install pip wheel setuptools --upgrade  
pip install yangsuite  
yangsuite
```

Install Pyenv on Ubuntu 22.04

Get the Required Dependencies

```
sudo apt-get update; sudo apt-get install make build-essential libssl-dev \  
zlib1g-dev \  
libbz2-dev libreadline-dev libsqlite3-dev wget curl llvm \  
libncursesw5-dev xz-utils tk-dev libxml2-dev libxmlsec1-dev libffi-dev liblzma-dev
```

Install Pyenv

<https://github.com/pyenv/pyenv-installer>

```
curl https://pyenv.run | bash
```

Make sure this is in your `.bashrc`

```
export PATH="$HOME/.pyenv/bin:$PATH"  
eval "$(pyenv init --path)"  
eval "$(pyenv virtualenv-init -)"
```

```
pyenv install 3.10.6
```

Install options: docker and pip

Docker:

Git clone

Run a script

3 docker container solution

HTTPS support

Using docker-compose

Default port: <https://localhost:8443>

Pip:

Install Python, pip, venv

Pip install YANG Suite

Requires python 3.8

Virtual Environment

Direct HTTP access into the tool only

Default port: <http://localhost:8480>

The screenshot shows the web interface of the Cisco YANG Suite. The URL <https://localhost:8443> is highlighted in a light blue box. The page title is "Cisco YANG Suite!". A dark blue sidebar on the left contains navigation links: Setup, Explore, Protocols, and Help. The main content area has a heading "Welcome to Cisco YANG Suite!" followed by a paragraph: "YANG Suite is a set of tools related to YANG models (RFC 6020, RFC 7950) and related technologies such as NETCONF (RFC 6241). It provides a modular infrastructure which various YANG application plugins can be used. Please check developer.cisco.com/yangsuite for information, learning labs, and announcements." Below this is a section titled "Docker-based Installation" with a sub-heading "The yangsuite/docker/start_yang_suite.sh script performs the following:" and a bulleted list: "Prompts for username, password, and email which will be the superuser to yangsuite.", "Gives the choice of creating test SSL/TLS certificate and key.", "Creates an environment file needed for the yangsuite docker container install.", and "Runs docker-compose up."

The screenshot shows the web interface of the Cisco YANG Suite. The URL <http://localhost:8480> is highlighted in a light blue box. The page title is "Cisco YANG Suite!". A dark blue sidebar on the left contains navigation links: Setup, Explore, Protocols, and Help. The main content area has a heading "Welcome to Cisco YANG Suite!" followed by a paragraph: "YANG Suite is a set of tools related to YANG models (RFC 6020, RFC 7950) and related technologies such as NETCONF (RFC 6241). It provides a modular infrastructure which various YANG application plugins can be used. Please check developer.cisco.com/yangsuite for information, learning labs, and announcements." Below this is a section titled "Docker-based Installation" with a sub-heading "The yangsuite/docker/start_yang_suite.sh script performs the following:" and a bulleted list: "Prompts for username, password, and email which will be the superuser to yangsuite.", "Gives the choice of creating test SSL/TLS certificate and key.", "Creates an environment file needed for the yangsuite docker container install.", and "Runs docker-compose up."

Persistent YANG Suite with Linux Screen Sessions

Goal: start pip yangsuite automatically on reboot for a persistent YANG Suite (no need to restart or reinstall).

Use the Linux "screen" function and add some additional syntax to run it as a cronjob.

<https://linuxize.com/post/how-to-use-linux-screen/>

```
#Yangsuite autostart sequence

sudo cat <<EOF >>/home/cisco/.clus_startup.sh
cd ~/testing/YANGSuite
source .venv/bin/activate
yangsuite
EOF

crontab -e
#add this line to the file after entering the "crontab -e" command
@reboot /usr/bin/screen -dm bash -c 'sleep 5; /home/cisco/.clus_startup.sh; exec sh'
```

Capabilities & Examples

- Device Management
- YANG Management

Accessing YANG Suite HTTP User Interface

Access YANG Suite using `https://localhost`

Login using the credentials as provided in `docker-compose.yml`

Log in to YANG Suite

Please login to access this page.

Username:

Password:

Login

[Lost your password?](#)

Add a Device

- Setup > Device profiles > Create new device
- NETCONF Timeout is set here, default 30 seconds and can be increased to 300s as needed

New Device Profile

Fields marked with * are required.

General Info

Profile Name *	<input type="text"/>				
Description	<input type="text"/>				
Address *	<input type="text"/>				
Username	<input type="text"/>				
Password	<input type="text"/>				
Timeout *	<input type="text" value="30"/>				
Variables	<table><tr><th>Variable Name</th><th>Value</th></tr><tr><td><input type="text"/></td><td><input type="text"/></td></tr></table>	Variable Name	Value	<input type="text"/>	<input type="text"/>
Variable Name	Value				
<input type="text"/>	<input type="text"/>				

NETCONF

<input type="checkbox"/> Device supports NETCONF	
Device Variant *	(Default - RFC-compliant device)
NETCONF port *	830
<input type="checkbox"/> Skip SSH key validation for this device	
Address	(same as base)
Username	(same as base)
Password	(same as base)
Timeout	30

TLS Authority Certificate:	<input type="text" value="Choose File no file selected"/>
TLS Client Certificate:	<input type="text" value="Choose File no file selected"/>
TLS Client Key:	<input type="text" value="Choose File no file selected"/>
TLS Server Certificate:	<input type="text" value="Choose File no file selected"/>
TLS Server Key:	<input type="text" value="Choose File no file selected"/>

RESTCONF

<input type="checkbox"/> Device supports RESTCONF	
HTTP or HTTP(secure) encoding	https
RESTCONF base URL	/restconf
RESTCONF port *	443
Address	(same as base)
Username	(same as base)
Password	(same as base)

gNMI

<input type="checkbox"/> Device supports gNMI	
Platform *	IOS XE
gNMI insecure port *	50052
gNMI secure port *	9339
<input type="checkbox"/> Use TLS Certificate	
TLS host override	<input type="text"/>

Check Device Connectivity

The screenshot displays the Cisco YANG Suite interface. On the left is a blue navigation sidebar with the following menu items: Admin, Setup, YANG files and repositories, YANG module sets, Device profiles, Analytics, Explore, Protocols, Test Manager, and Help. The main content area is titled "Manage device profiles" and includes a "Select a device profile" section with a radio button for "C9300". Below this are several action buttons: "Create new device", "Check selected device's reachability", "Clone selected device", "Edit selected device", "Delete selected device", and "Create default Repository and Yangset". A modal popup titled "Connectivity check results for 'c930...'" is open, showing a list of checked items: ping, gNMI, NETCONF, and RESTCONF.

Cisco YANG Suite

YANG Suite / Device profiles

Manage device profiles

admin

Select a device profile

C9300

Create new device

Check selected device's reachability

Clone selected device

Edit selected device

Delete selected device

Create default Repository and Yangset

Connectivity check results for "c930..."

Connectivity check results:

- ✓ ping
- ✓ gNMI
- ✓ NETCONF
- ✓ RESTCONF

Add YANG Modules into the local repository Default Repository and Yangset

Cisco YANG Suite

Admin Setup YANG files and repositories YANG module sets Device profiles Analytics Explore Protocols Test Manager Help

YANG Suite / Device profiles
Manage device profiles

Select a device profile
 C9300

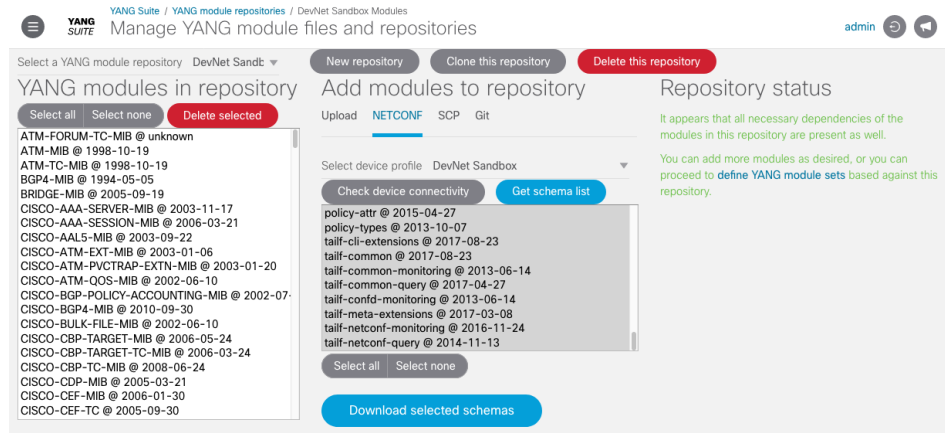
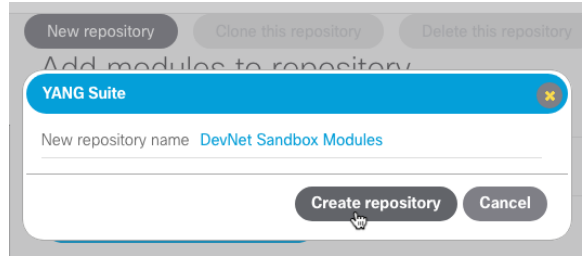
Create new device
Check selected device's reachability
Clone selected device
Edit selected device
Delete selected device
Create default Repository and Yangset

Add YANG Modules using NETCONF

Create YANG Repository

NETCONF > Select device
> Get schema list > Select
all > Download

All data models will now
be downloaded from the
NETCONF interface into
the local YANG Suite



Add YANG Modules from local file upload

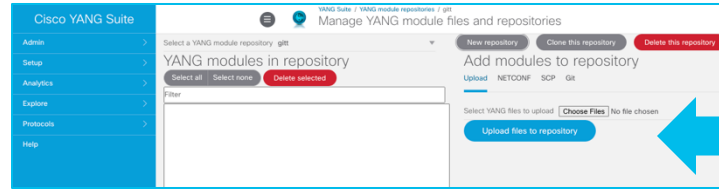
Select Setup > YANG Files and Repository

Create/Select YANG Repository

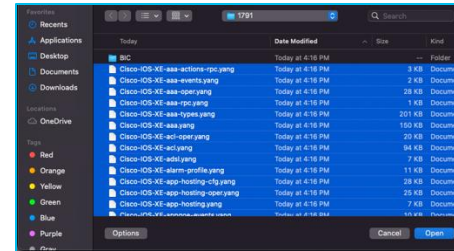
Navigate to “Upload” tab

Select the necessary YANG files.

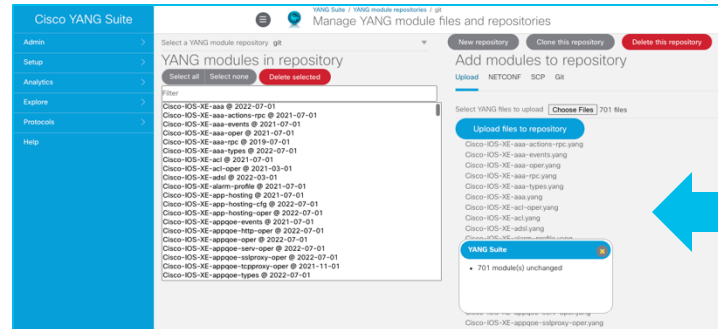
Select “Upload files to repository”.



Click “Upload files to repository” button



Select files to Upload



View newly uploaded modules

Add YANG modules to a working set

Filter for “Cisco-IOX-XE” native models

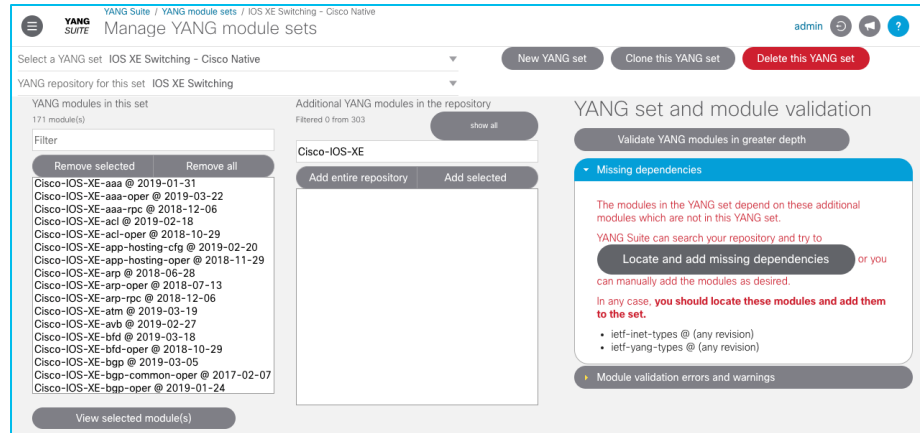
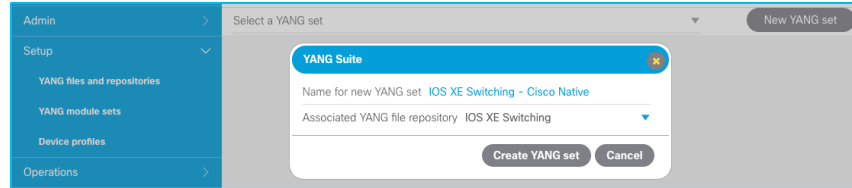
Select all > “Add selected”

Select “Locate and add missing dependencies”

Validation complete

This YANG set is currently empty.

Please add one or more YANG modules to this set.



Import YANG from Github

Easily import YANG from Github repository

New repository Clone this repository Delete this repository

Add modules to repository

Upload NETCONF SCP **Git**

Repository URL `https://github.com/openconfig/public/`

Git branch `master`

Directory within repository `release/models`

Include subdirectories

Import YANG files

Add modules to repository

Upload NETCONF SCP **Git**

Repository URL `https://github.com/YangModels/yang`

Git branch `main`

Directory within repository `vendor/cisco/xe/17131`

Include subdirectories

Import YANG files

- Explore

Explore YANG

Operations > Explore YANG

Select YANG Set and module then Load module

Details about the data are displayed

YANG Suite | Explore YANG Models | admin

Select a YANG set: C9300-17.1 | Select YANG module(s): Cisco-IOS-XE-interfaces-oper | Load module(s)

Icon legend | Search XPath(s) | Search nodes | Expand all nodes

Display schema nodes only | Display all nodes

Cisco-IOS-XE-interfaces-oper

- interfaces
 - interface
 - name
 - interface-type
 - admin-status
 - oper-status
 - last-change
 - if-index
 - phys-address
 - higher-layer-if
 - lower-layer-if
 - speed
 - statistics
 - diffserv-info
 - vrf
 - ipv4
 - ipv4-subnet-mask
 - description
 - mtu
 - input-security-acl
 - output-security-acl
 - v4-protocol-stats
 - v6-protocol-stats
 - bia-address
 - ipv6-addr
 - lgg-aggregate-state
 - ipv4-tcp-adjust-mss
 - ipv6-tcp-adjust-mss
 - interface-class-choice

Node Properties

Name: statistics
Node type: container
Description: A collection of interface-related statistics objects
Module: Cisco-IOS-XE-interfaces-oper
Revision: 2019-11-01
XPath: /interfaces/interface/statistics
Prefix: interfaces-ios-xe-oper
Namespace: http://cisco.com/ns/yang/Cisco-IOS-XE-interfaces-oper
Access: read-only
Operations: + "get"
Schema Node Id: /interfaces/interface/statistics
Reference URL: <https://tools.ietf.org/html/rfc6020#section-7.5>
7.5. The container Statement

The "container" statement is used to define an interior data node in the schema tree. It takes one argument, which is an identifier, followed by a block of substatements that holds detailed container information.

A container node does not have a value, but it has a list of child nodes in the data tree. The child nodes are defined in the container's substatements.

RFC 6020 | Standards Track | YANG | October 2010

Tree Icon Legend

Node Icons

action

anydata

anyxml

case

choice

container

input

leaf

leaf-list

list

module

notification

output

rpc

submodule

Node Support

status: deprecated

status: obsolete

~~deviation: not supported~~

Node Badges

list key

Demo: Explore YANG

Explore YANG model tree and container elements in detail

The screenshot displays the YANG Suite interface for exploring YANG models. The top navigation bar shows the current context: 'C9300 - 17.2' and selected modules 'openconfig-interfaces' and 'Cisco-IOS-XE-process-cpu-oper'. The main area is split into two panes. The left pane shows a hierarchical tree of the YANG model, with 'five-seconds' selected under 'cpu-usage/cpu-utilization'. The right pane, titled 'Node Properties', provides detailed information for the selected node.

Property	Value
Name	five-seconds
Nodetype	leaf
Datatype	uint8
Description	Busy percentage in last 5-seconds
Module	Cisco-IOS-XE-process-cpu-oper
Revision	2019-05-01
Xpath	/cpu-usage/cpu-utilization/five-seconds
Prefix	process-cpu-ios-xe-oper
Namespace	http://cisco.com/ns/yang/Cisco-IOS-XE-process-cpu-oper
Units	percent
Min	0
Max	255
Access	read-only
Operations	<ul style="list-style-type: none">"get"
Schema Node Id	/cpu-usage/cpu-utilization/five-seconds

Reference URL:
<https://tools.ietf.org/html/rfc6020#section-7.6>

7.6. The leaf Statement

The "leaf" statement is used to define a leaf node in the schema tree. It takes one argument, which is an identifier, followed by a block of substatements that holds detailed leaf information.

A leaf node has a value, but no child nodes in the data tree. Conceptually, the value in the data tree is always in the canonical form (see Section 9.1).

A leaf node exists in zero or one instances in the data tree.

The "leaf" statement is used to define a scalar variable of a particular built-in or derived type.

Explore YANG Models for interfaces-oper

Cisco YANG Suite

YANG Suite / Exploring YANG / YANG set "c9300-default-yangset" / Modules

Explore YANG Models

Select a YANG set: c9300-default-yangset

Select YANG module(s): Cisco-IOS-XE-interfaces-oper

Load module(s)

Icon legend | Search XPath(s) | Search nodes | Expand all nodes

Display schema nodes only | Display all nodes

Node Properties

Name	interface-type
NodeType	leaf
Datatype	interfaces-ios-xe-oper:ietf-intf-type
Basetype	enumeration
Description	When an interface entry is created, a server MAY initialize the type leaf with a valid value, e.g., if it is possible to derive the type from the name of the interface. If a client tries to set the type of an interface to a value that can never be used by the system, e.g., if the type is not supported or if the type does not match the name of the interface, the server MUST reject the request. A NETCONF server MUST reply with an rpc-error with the error-tag 'invalid-value' in this case
Module	Cisco-IOS-XE-interfaces-oper
Revision	2021-03-01
Xpath	/interfaces/interface/interface-type
Prefix	interfaces-ios-xe-oper
Namespace	http://cisco.com/ns/yang/Cisco-IOS-XE-interfaces-oper
Schema Node Id	/interfaces/interface/interface-type
Access	read-only
Operations	<ul style="list-style-type: none">"get"

Reference URL:
<https://tools.ietf.org/html/rfc6020#section-7.6>

7.6. The leaf Statement

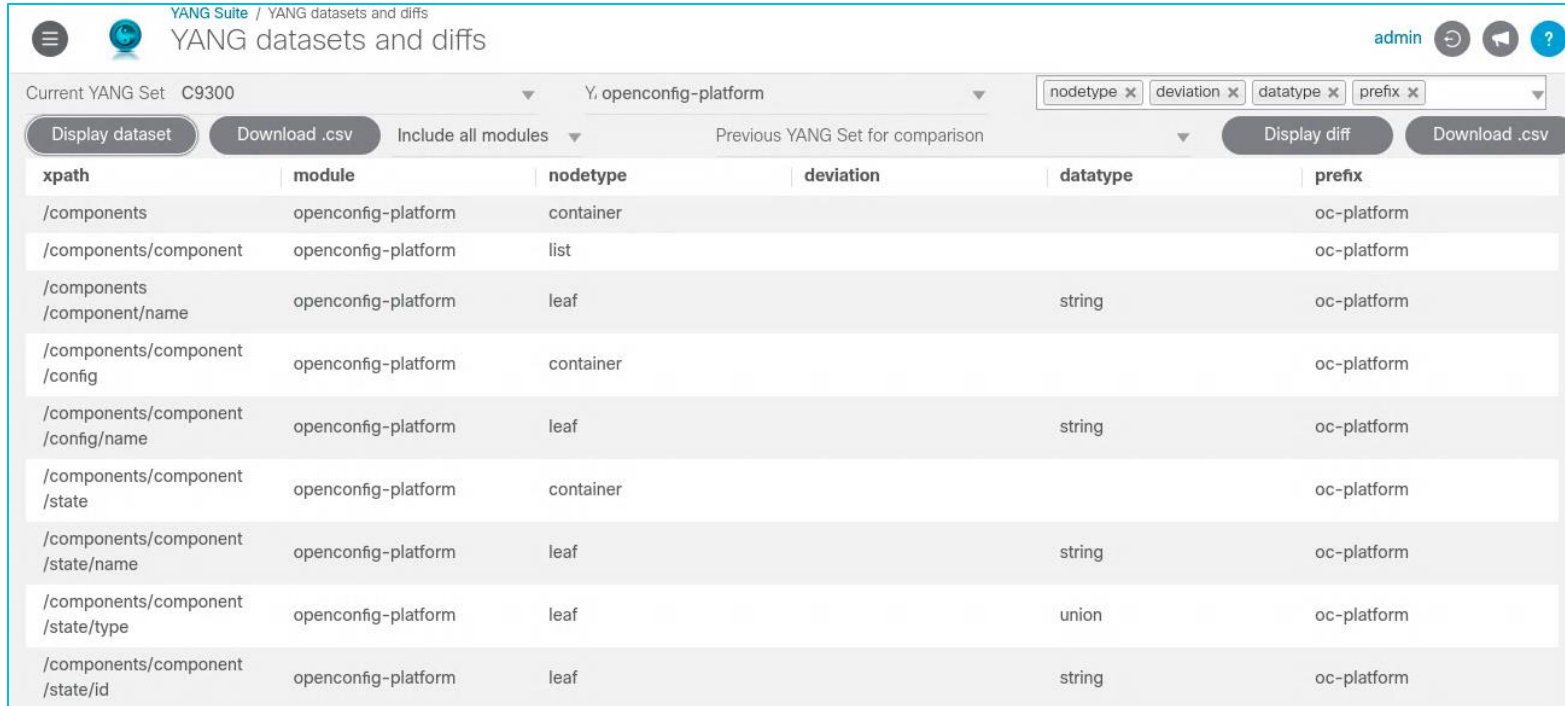
The "leaf" statement is used to define a leaf node in the schema tree. It takes one argument, which is an identifier, followed by a block of substatements that holds detailed leaf information.

A leaf node has a value, but no child nodes in the data tree. Conceptually, the value in the data tree is always in the canonical form. (see Section 8.1)

- Datasets & Diffs

Demo: Datasets and Diffs

Compare YANG, run diffs, display and export metadata to view see changes for various software releases



The screenshot displays the 'YANG datasets and diffs' interface. At the top, it shows the current YANG Set as 'C9300' and the selected module as 'Y. openconfig-platform'. Below this, there are buttons for 'Display dataset', 'Download .csv', and 'Include all modules'. A dropdown menu for 'Previous YANG Set for comparison' is also visible. The main content is a table with columns for 'xpath', 'module', 'nodetype', 'deviation', 'datatype', and 'prefix'. The table lists various YANG nodes and their associated metadata.

xpath	module	nodetype	deviation	datatype	prefix
/components	openconfig-platform	container			oc-platform
/components/component	openconfig-platform	list			oc-platform
/components/component/name	openconfig-platform	leaf		string	oc-platform
/components/component/config	openconfig-platform	container			oc-platform
/components/component/config/name	openconfig-platform	leaf		string	oc-platform
/components/component/state	openconfig-platform	container			oc-platform
/components/component/state/name	openconfig-platform	leaf		string	oc-platform
/components/component/state/type	openconfig-platform	leaf		union	oc-platform
/components/component/state/id	openconfig-platform	leaf		string	oc-platform

Demo: Using YANG Suite to compare versions

Using the Analytics > Datasets and Diff's plugin to YANG Suite
The deviation for the “partner-part-num” is no longer present

Support for telemetry with subscribe to LACP operational data

The screenshot displays the Cisco YANG Suite interface. The left sidebar contains navigation options: Admin, Setup, Analytics, Datasets and diffs (selected), SNMP to YANG Mapping, YANG coverage, Explore, Protocols, Test Manager, and Help. The main content area shows the 'YANG datasets and diffs' section. It includes a search bar for 'Current YANG Set' (1711-c9300) and 'Previous YANG Set for comparison' (jchoae-c9300-2-default-yangset). Below the search bar are buttons for 'Display dataset', 'Download .csv', and 'Include all modules'. A table displays the comparison results:

?	xpath	deviation	description	module
<	/lacp/interfaces/interface/members/member/state/partner-part-num	not-supported	Port number of the partner (remote) port for this member port	openconfig-lacp
>	/lacp/interfaces/interface/members/member/state/partner-part-num		Port number of the partner (remote) port for this member port	openconfig-lacp

Understanding Data Set Diff

Current YANG Set GithubOC YANG module openconfig-interfaces deviation x description x

Display dataset Download .csv Include all modules Previous YANG Set for comparison 1711-c9300

Display diff Download .csv

? xpath	module	deviation	description
< /interfaces/interface/name	openconfig-interfaces		References the configured name of the interface
> /interfaces/interface/name	openconfig-interfaces		References the name of the interface

Legend:

- > Current YANG set
- < Previous YANG set
- + In OC only

Select a YANG set GithubOC Load module(s)

Select YANG module(s)

Icon legend Search XPath(s) Search nodes Expand all nodes

Display schema nodes only Display all nodes

openconfig-interfaces

- interfaces
 - interface
 - name
 - config
 - state

Name	Value
Name	name
Nodetype	leaf
Datatype	leafref
Description	References the name of the interface

- NETCONF

NETCONF

Protocols > NETCONF

Select YANG Set

Select Module: Cisco-IOS-XE-environment-oper

Load Module

NETCONF Operation: get

Device: select device

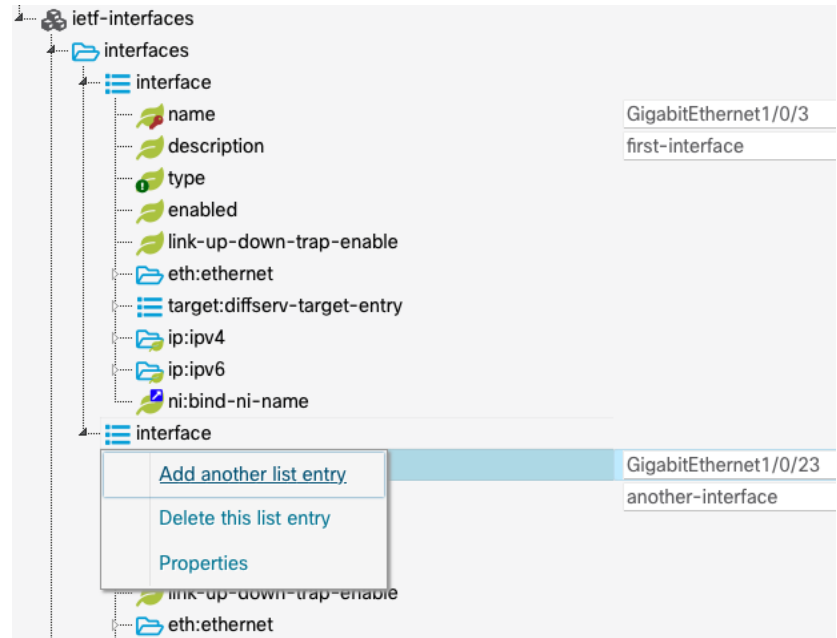
The screenshot displays the YANG Suite interface. The top navigation bar shows the path: YANG Suite / NETCONF / YANG set "IOS XE Switching - Cisco Native" / Modules. The main interface includes a sidebar on the left with a menu for "YANG Suite" containing options like Admin, Setup, Operations, Analytics, Mapper, Protocols, gNMI, gRPC telemetry, NETCONF, and RESTCONF. The main content area has a header "NETCONF" and a breadcrumb "YANG Suite / NETCONF / YANG set 'IOS XE Switching - Cisco Native' / Modules". Below this, there are several controls: "YANG Set" (IOS XE Switching - Cisco Native), "Module(s)" (Cisco-IOS-XE-environment-oper), "NETCONF Operation" (get), and "Device" (jcohoe-cat9300). There are buttons for "Load Module(s)", "Edit...", "Open Device Window", "YANG Tree", "Replays", "RPC Options...", "Build RPC", "Run RPC(s)", and "Clear RPC(s)". A table with columns "Nodes", "Value", and "Operation" is visible, showing a single entry for "Cisco-IOS-XE-environment-oper".

Configure multiple interfaces in a single payload

The usecase to modify multiple interfaces within a single NETCONF session is supported

Additional lists/interfaces can be set by by selecting “Add another list entry” after right-click the “interface” list

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name>GigabitEthernet1/0/3</name>
          <description>first-interface</description>
        </interface>
        <interface>
          <name>GigabitEthernet1/0/23</name>
          <description>another-interface</description>
        </interface>
      </interfaces>
    </config>
  </edit-config>
</rpc>
```



```
VNC2-LEAF4#sh int status | i 3
Gi1/0/3   first-interface   connected  1
Gi1/0/13  WIRELESS-AP         disabled   1
Gi1/0/23  another-interface   notconnect 1
Te1/1/3   notconnect           notconnect 1
```

NETCONF get environment example

This example shows the NETCONF get operation using the Cisco-IOS-XE-environment-oper YANG model reports the “Inlet Temp” is 40 Celsius

The screenshot shows the YANG Suite interface for a NETCONF operation. The top navigation bar includes 'YANG Suite', 'YANG SUITE', 'NETCONF', and 'YANG set "IOS XE Switching - Cisco Native" / Modules'. The main area is divided into three sections:

- Nodes:** A tree view showing the hierarchy of the YANG model. The selected node is 'environment-sensor'.
- Value:** A table listing the values for the selected node and its children.
- Operation:** A table showing the operation being performed on the selected node.

The 'Value' table contains the following data:

Node	Value
environment-sensor	
name	
location	
state	
current-reading	
sensor-units	
low-critical-threshold	
low-normal-threshold	
high-normal-threshold	
high-critical-threshold	
sensor-name	

The 'Operation' table shows the operation being performed on the selected node:

Operation
get

The terminal window shows the XML response from the device:

```
Sendings:
#324
<ncrpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:5d174829-1847-4104-b01f-f1b22ac09f4">
  <nciget>
    <filter>
      <environment-sensors xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-environment-oper">
        <environment-sensor/>
      </environment-sensors>
    </filter>
  </nciget>
</ncrpc>

Received:
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:5d174829-1847-4104-b01f-f1b22ac09f4">
  <data>
    <environment-sensors xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-environment-oper">
      <environment-sensor>
        <name>Inlet Temp Sena</name>
        <location>Switoh 1</location>
        <state>ONEN</state>
        <current-reading>40</current-reading>
        <sensor-units>Celsius</sensor-units>
        <low-critical-threshold>0</low-critical-threshold>
        <low-normal-threshold>0</low-normal-threshold>
        <high-normal-threshold>0</high-normal-threshold>
        <high-critical-threshold>0</high-critical-threshold>
      </environment-sensor>
    </environment-sensors>
  </data>
</rpc-reply>
```

Build NETCONF XML Payload to GET Hostname

The screenshot displays the Cisco YANG Suite interface for configuring a NETCONF RPC. The interface includes a sidebar with navigation options (Admin, Setup, Analytics, Explore, Protocols, Test Manager, Help) and a main workspace. The workspace shows the YANG Tree for the Cisco-IOS-XE-native module, with the 'hostname' node selected. The 'Value' column for this node contains the text 'string'. The 'NETCONF Operation' is set to 'get-config', and the 'Device' is 'C9300'. The 'Build RPC' button is highlighted, and the resulting XML payload is shown in the right-hand pane.

YANG Suite / NETCONF / YANG set "c9300-default-yangset" / Modules

admin

YANG Set: c9300-default-yangset | Module(s): Cisco-IOS-XE-native x | Load Module(s)

NETCONF Operation: get-config | Device: C9300 | Edit Device | Open Device Window

YANG Tree | Replays | RPC Options... | Build RPC | Run RPC(s) | Clear RPC(s)

Nodes | Value

Cisco-IOS-XE-native

- native
 - default
 - bfd
 - version
 - stackwise-virtual
 - boot-start-marker
 - boot
 - boot-end-marker
 - banner
 - captive-portal-bypass
 - memory
 - location
 - call-home
 - hw-module
 - cisp
 - module
 - domain
 - parser
 - service
 - platform
 - hostname: string
 - enable
 - password
 - eap
 - archive
 - username
 - user-name

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <hostname/>
      </native>
    </filter>
  </get-config>
</rpc>
```

Device Response after NETCONF GET

Start Session

Datstores: Candidate Running Startup Actions:

```
Sending:
#336
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:4c8cb5ac-15c6-4fae-83bc-e9f9b1f7ad43">
  <nc:get-config>
    <nc:source>
      <nc:running/>
    </nc:source>
    <nc:filter>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <hostname/>
      </native>
    </nc:filter>
  </nc:get-config>
</nc:rpc>

##

Received message from host

<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:4c8cb5ac-15c6-4fae-83bc-e9f9b1f7ad43" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
      <hostname>netconf-pro</hostname>
    </native>
  </data>
</rpc-reply>

NETCONF get-config COMPLETE

Requesting 'CloseSession'

Sending:
#184
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:bd42736b-d523-4308-9780-e491928d8b82">
  <nc:close-session/>
</nc:rpc>

##
```

Build NETCONF XML Payload to SET Hostname

The screenshot displays the Cisco YANG Suite interface for configuring a NETCONF RPC. The top navigation bar shows the current context: YANG Suite / NETCONF / YANG set "c9300-default-yangset" / Modules. The user is logged in as 'admin'. The main workspace is divided into three sections:

- Left Panel (Nodes):** A tree view of the YANG model for 'Cisco-IOS-XE-native'. The 'hostname' node is selected, and its value is set to 'netconf-pro'.
- Top Panel (Configuration):** Shows the YANG Set as 'c9300-default-yangset' and the Module(s) as 'Cisco-IOS-XE-native'. The NETCONF Operation is set to 'edit-config' and the Device is 'C9300'. Buttons for 'Edit Device', 'Run RPC(s)', and 'Clear RPC(s)' are visible.
- Right Panel (XML Payload):** Displays the generated NETCONF XML payload for the selected configuration:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <hostname>netconf-pro</hostname>
      </native>
    </config>
  </edit-config>
</rpc>
```

Device Response after NETCONF SET

Start Session

Datstores:

 Candidate

 Running

 Startup

Actions:



Waiting for Data...

Sending:

```
#359
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:a4a94a42-381b-4057-83c1-6d97035cbbcd">
  <nc:edit-config>
    <nc:target>
      <nc:running/>
    </nc:target>
    <nc:config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <hostname>netconf-pro</hostname>
      </native>
    </nc:config>
  </nc:edit-config>
</nc:rpc>

##
```

Received message from host

```
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:a4a94a42-381b-4057-83c1-6d97035cbbcd" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

NETCONF edit-config COMPLETE

Requesting 'CloseSession'

Sending:

```
#184
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:6c624a67-0d7a-4833-847c-9028c4fb4e92">
  <nc:close-session/>
</nc:rpc>

##
```

Received message from host

Generate Python Script



YANG Suite / NETCONF / YANG set "c9300-default-yangset" / Modules

NETCONF

YANG Set c9300-default-yangset

Module(s)

Cisco-IOS-XE-native x

NETCONF Operation

get-config

Device

C9300



Edit Device

Open Device Window



YANG Tree



Replays

RPC Options...



Build RPC



Run RPC(s)

Nodes

- Cisco-IOS-XE-
 - native
 - default
 - bfd
 - version
 - stackwi
 - boot-st
 - boot

- Load Replay...
- Save as New Replay...
- Edit Replay Information...
- Save Changes
- Generate Python script

Value

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf"
  <get-config>
  <source>
    <running/>
  </source>
  <filter>
    <native xmlns="http://cisco.com/ns/yang"
      <hostname/>
    </native>
  </filter>
</get-config>
</rpc>
```

Demo: Generate Python from NETCONF

Access YANG Suite >
Protocols > NETCONF

Select YANG model then
Build RPC payload

Select Replays

Select Generate Python
script

The screenshot displays the Cisco YANG Suite interface. On the left, a sidebar contains navigation options: Admin, Setup, Explore, Protocols, and Help. The main area is titled 'NETCONF' and shows the YANG Set 'c9300-default-yangset' and the selected Module 'Cisco-IOS-XE-native'. The 'NETCONF Operation' dropdown is set to 'edit-config' for Device 'C9300'. A 'Replays' menu is open, with 'Generate Python script' highlighted. The 'Nodes' tree on the left shows a hierarchy including 'ios-vlan:access-log', 'ios-vlan:group', 'ios-vlan:list', and 'ios-vlan:id'. The right pane shows the XML payload for the RPC, and the bottom pane displays the generated Python script. The script includes imports for 'xml.etree as et', 'argparse import ArgumentParser', 'ncclient import manager', and 'ncclient.operations import RPCError'. It defines a 'payload' dictionary and a 'parser' object with arguments for '-a', '-u', '-p', and '-port'. The script ends with 'args = parser.parse_args()'.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <vlan>
          <id>28</id>
        </vlan>
      </native>
    </config>
  </edit-conf>
</rpc>
```

```
1 #! /usr/bin/env python
2 import xml.etree as et
3 from argparse import ArgumentParser
4 from ncclient import manager
5 from ncclient.operations import RPCError
6
7 payload = [
8     ...
9 <get xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
10 <filter>
11 <lag-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-lacp-o
12 </filter>
13 </get>
14 ...
15 ]
16
17 if __name__ == '__main__':
18
19     parser = ArgumentParser(description='Usage:')
20
21     # script arguments
22     parser.add_argument('-a', '--host', type=str, required=True,
23                         help="Device IP address or Hostname")
24     parser.add_argument('-u', '--username', type=str, required=True,
25                         help="Device Username (netconf agent username)")
26     parser.add_argument('-p', '--password', type=str, required=True,
27                         help="Device Password (netconf agent password)")
28     parser.add_argument('--port', type=int, default=830,
29                         help="Netconf agent port")
30     args = parser.parse_args()
```

YANG Suite + Ansible

using NETCONF, RESTCONF & gNMI OpenConfig

Cisco YANG Suite

- Admin
- Setup
- Analytics
 - Datasets and diffs
 - SNMP to YANG Mapping
 - YANG coverage
- Explore
 - YANG
- Protocols
 - gNMI
 - gRPC telemetry
 - NETCONF**
 - RESTCONF
- Test Manager
- Help

Quickly and easily generate Ansible playbook for deployments to be used with the inventory, similar to the “Generate Python script” button.

The screenshot shows the Cisco YANG Suite interface. The 'NETCONF' option in the Protocols sidebar is highlighted with a yellow box. In the main control area, the 'Generate Ansible playbook' button is also highlighted with a yellow box. A dialog box titled 'Ansible Playbook Settings' is open, showing fields for 'Ansible file name' (ansible.yaml), 'Ansible playbook name' (NETCONF playbook), and 'Ansible task name' (NETCONF RPC). Below the dialog, a preview of the generated Ansible playbook is shown, including host definitions, connection settings, and a task to establish a subscription.

```
1 ---
2 -- hosts: ios-xe
3 gather_facts: no
4 connection: netconf
5 remote_user: admin
6
7 tasks:
8 - name: establish subscription
9   netconf_config:
10     xml: |
11       <nc:config xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
12         <mt-config-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-cfg">
13           <mt-subscription>
14             <subscription-id>501</subscription-id>
15             <base>
16               <stream-yang-push/stream>
17                 <encoding-encode-kvpb/>
18                 <source-address>10.0.0.19</source-address>
19                 <source-vrf>mgmt-vrf</source-vrf>
20               <period>3000</period>
21             <path>/process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds</xpath>
```

Restrictions for gNMI

For OpenConfig YANG only

Using 3rd party plugin to Ansible from Nokia, which requires collection to be installed:

```
$ ansible-galaxy collection install nokia.grpc
```

NETCONF + Ansible

Similar to the “Generate Python” functionality, a new functionality generates YAML formatted for Ansible.

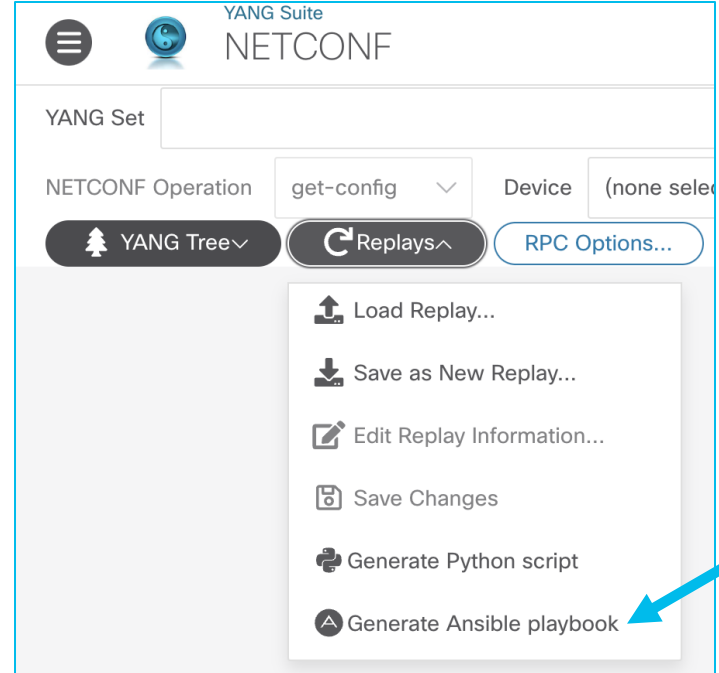
Requirements

1. Install Ansible
2. Install NETCONF collection:

```
ansible-galaxy collection install NETCONF
```

```
- name : conf-host
  hosts: c9300
  connection: netconf
  gather_facts: no

  tasks:
    - name: hostname-conf
      netconf_config:
        xml: |
          <config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
            <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
              <hostname>c9300-pod29</hostname>
            </native>
          </config>
```



The screenshot shows the YANG Suite NETCONF web interface. At the top, there is a navigation bar with a hamburger menu icon, a globe icon, and the text 'YANG Suite NETCONF'. Below this, there are several controls: 'YANG Set' (a text input field), 'NETCONF Operation' (a dropdown menu currently showing 'get-config'), and 'Device' (a dropdown menu currently showing '(none selected)'). Below these are three buttons: 'YANG Tree' (with a tree icon), 'Replays' (with a refresh icon), and 'RPC Options...' (with a gear icon). The 'Replays' button is active, and its dropdown menu is open, showing several options: 'Load Replay...', 'Save as New Replay...', 'Edit Replay Information...', 'Save Changes', 'Generate Python script', and 'Generate Ansible playbook'. A blue arrow points to the 'Generate Ansible playbook' option.

NETCONF + Ansible Change Host Name demo

The screenshot shows the Cisco YANG Suite web interface. On the left is a blue navigation sidebar with the following menu items: Admin, Setup, Analytics, Explore, Protocols, Test Manager, and Help. The main content area is titled "Welcome to Cisco YANG Suite!" and contains the following text:

YANG Suite is a set of tools related to YANG models (RFC 6020, RFC 7950) and related technologies such as NETCONF (RFC 6241).

It provides a modular infrastructure which various YANG application plugins can be used.

Please check developer.cisco.com/yangsuite for information, learning labs, and announcements.

Docker-based Installation

The `yangsuite/docker/start_yang_suite.sh` script performs the following:

- Prompts for username, password, and email which will be the superuser to yangsuite.
- Gives the choice of creating test SSL/TLS certificate and key.
- Creates an environment file needed for the yangsuite docker container install.
- Runs `docker-compose up`.

On the right side of the interface, there is a "YANG Suite Documentation" section with a search bar and a list of links:

- Welcome to Cisco YANG Suite!
- Device Profiles
 - Managing device profiles
 - Setting up YANG Suite TLS/SSL
- File Manager
 - Constructing and populating a YANG module repository
 - Uploading YANG files from the local filesystem to a YANG repository
 - Downloading YANG files via NETCONF from a device to a YANG repository
 - Copying YANG modules via SCP from a server to a YANG repository
 - Importing YANG modules from a Git repository to a YANG repository
 - Defining a YANG module set
 - Managing YANG module files in YANG Suite
- Working with YANG Models
 - Exploring YANG Models
- YANG Suite Analytics
 - Datasets
 - Mapping SNMP OIDs to YANG XPathS
- Using gNMI with YANG Suite
- YANG Suite gRPC telemetry receivers
 - Telemetry over gRPC Clear Channel
 - Configuring gRPC Telemetry Receivers
 - Telemetry over gRPC Secure Channel
- Using NETCONF with YANG Suite
 - Using NETCONF RPCs
 - Working with NETCONF Notification Streams
 - Using Replays for repeated workflows
 - Locking and unlocking datastores
- RESTCONF in YANG Suite
 - YANG Suite RESTCONF
- Test management with YANG Suite
 - Using Test Manager to define and execute tests
 - Convert Ytool Test Suites to YANG Suite Format
 - Generating YANG model Tests
 - Importing and exporting YANG Suite tests
 - Managing and editing replay files
 - Export YANG Suite tests pyATS Format

NETCONF + Ansible Update Interface Description Demo

The screenshot displays the Cisco YANG Suite web interface. On the left is a blue navigation sidebar with the following items: Admin, Setup, Analytics, Explore, Protocols, Test Manager, and Help. The main content area features a header with 'Welcome to Cisco YANG Suite!' and a sub-header 'Help: Welcome to Cisco YANG Suite!'. Below this, there is a paragraph describing the suite as a set of tools for YANG models and related technologies like NETCONF. A section titled 'Docker-based Installation' follows, explaining that a specific script performs several tasks: prompting for user credentials, offering to create test certificates, setting up the docker environment, and running docker-compose. On the right, a sidebar titled 'YANG Suite Documentation' includes a search bar and a comprehensive list of topics such as 'Welcome to Cisco YANG Suite!', 'Device Profiles', 'File Manager', 'Working with YANG Models', and 'Test management with YANG Suite'.

Welcome to YANG Suite!

YANG Suite is a set of tools related to YANG models (RFC 6020, RFC 7950) and related technologies such as NETCONF (RFC 6241). It provides a modular infrastructure atop which various application plugins can be used.

Please subscribe to yang-suite-users@cisco.com for announcements.

Getting Started

If this is your first time running YANG Suite, you have two ways to begin:

1. Configure YANG Suite to **connect to a device** (router or switch) and then **download supported YANG models from the device**.
2. **Upload YANG model files** from your computer to YANG Suite.

Once you have provided YANG models to YANG Suite by either of the above methods, you can then proceed to **define sets of YANG models** of interest.

Once this is done you can proceed to learn how to **explore the contents of these YANG models** or (if yangsuite-netconf is installed) **use these models with NETCONF**.

For more information on any of these topics, use the table of contents or the search box, both on the right side of this page. To return to this page at any time, use the "Help" menu in the sidebar to the left of any YANG Suite page. For page-specific help, click the (?) icon in the top right corner.

About YANG Suite

The core application plugins for YANG Suite are:

[yangsuite-filemanager](#)

Manages storage of YANG schema files within the application. Allows users to upload YANG schema files to the application.

[yangsuite-devices](#)

Manages device profile definitions and access to devices.

[yangsuite-yangtree](#)

Provides basic browsing of YANG models in a tree format.

Additional (optional) application plugins include:

YANG Suite Documentation

Search docs

- **Welcome to YANG Suite!**
- **Device Profiles**
 - Managing device profiles
 - Setting Device to EnXR
- **Managing YANG module files in YANG Suite**
 - Constructing and populating a YANG module repository
 - Uploading YANG files from the local filesystem to a YANG repository
 - Downloading YANG files via NETCONF from a device to a YANG repository
 - Copying YANG modules via SCP from a server to a YANG repository
 - Importing YANG modules from a Git repository to a YANG repository
 - Defining a YANG module set
- **Working with YANG Models**
 - Exploring YANG Models
 - Generating collections of tests from a YANG model
 - Managing and editing replay files
- **YANG Suite Analytics user documentation**
 - Checking YANG model coverage
 - Datasets
- **Using gNMI with YANG Suite**
- **Receiving gRPC telemetry with YANG Suite**
- **Mapping YANG models**
 - Managing mapping projects
- **Using NETCONF with YANG Suite**
 - Using NETCONF RPCs
 - Working with NETCONF Notification Streams
 - Using Replays for repeated workflows
 - Locking and unlocking datastores
 - Setting Device to ENXR
- **RESTCONF in YANG Suite**
 - YANG Suite RESTCONF
- **Test management with YANG Suite**
 - Using Test Manager to define and execute tests
 - Convert Ytool Test Suites to YANG Suite Format
 - Generating tests based on autogenerated replays
 - Importing and exporting YANG Suite tests
 - Export YANG Suite tests in pyATS Format

- NETCONF + Telemetry

Demo: Model Driven Telemetry - NETCONF

Receive CPU telemetry every 10 seconds from NETCONF yang-push

The screenshot shows the NETCONF interface with the following configuration details:

- YANG Set: C9300
- Module(s): Cisco-IOS-XE-Interfaces-oper x | ietf-event-notifications x
- NETCONF Operation: (other RPC)
- Device: C9300
- Buttons: Edit Device, Open Device Window, Build RPC, Run RPC(s), Clear RPC(s)
- Nodes tree: Cisco-IOS-XE-Interfaces-oper > ietf-event-notifications > establish-subscription > input > stream > encoding > filter-type > rfc5277 > filter > yp:update-filter > yp:update-filter > yp:subtree > yp:xpath > yp:xpath-filter > cyp:native-filter > by-reference > yp:update-trigger > yp:periodic > yp:period (set to 1000) > yp:anchor-time > yp:on-change
- Value field: vp.yang-push
- Encoding: cyp.encode-kvqb

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <establish-subscription xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications">
    <stream xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push" yp:yang-push">
      <encoding xmlns:cyp="urn:cisco:params:xml:ns:yang:cisco-xe-ietf-yang-push-ext" cyp:encode-kvqb/>
      <xpath-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push"/>
      <period xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">1000</period>
    </establish-subscription>
  </rpc>
```

The first screenshot shows a received message from the host:

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2021-02-02T06:04:57.59Z</eventTime>
  <push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <subscription-id>2147483649</subscription-id>
    <datastore-contents-xml>
      <cpu-usage xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-process-cpu-oper">
        <cpu-utilization>
          <five-seconds>0</five-seconds>
        </cpu-utilization>
      </cpu-usage>
    </datastore-contents-xml>
  </push-update>
</notification>
```

The second screenshot shows a received message from the host with a highlighted section:

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2021-02-02T06:05:07.59Z</eventTime>
  <push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <subscription-id>2147483649</subscription-id>
    <datastore-contents-xml>
      <cpu-usage xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-process-cpu-oper">
        <cpu-utilization>
          <five-seconds>0</five-seconds>
        </cpu-utilization>
      </cpu-usage>
    </datastore-contents-xml>
  </push-update>
</notification>
```

* Note: ensure NETCONF session is not closed after sending the RPC by manually selecting "Start Session"

RFC5277 Event Streams

The RFC5277 event stream subscription is supported with the “get streams” and “subscribe” actions
These actions are accessed once the NETCONF session window is open and the session is connected
IOS XE supports both **NETCONF** and **snmpevents** streams

The NETCONF stream supports streaming of configuration change data

The snmpevents stream supports streaming of SNMP traps data

The screenshot shows the YANG Suite NETCONF interface. The top bar includes the YANG Set, Module(s) filter, and NETCONF Operation (set to 'get-config'). The Device is set to 'c9300'. The 'Open Device Window' button is highlighted with a blue box and an arrow pointing to the XML callout. The 'Actions' menu is open, showing the 'get streams' action highlighted with a blue box and an arrow pointing to the XML callout. The XML callout displays the following response:

```
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:8869c95c-1bd3-4a6b-b281-d7f1e269c03c"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:pa
<data>
  <netconf xmlns="urn:ietf:params:xml:ns:netmod:notification">
    <streams>
      <stream>
        <name>NETCONF</name>
        <description>default NETCONF event stream</description>
        <replySupport>false</replySupport>
      </stream>
      <stream>
        <name>snmpevents</name>
        <description>SNMP related notifications</description>
        <replySupport>true</replySupport>
        <replyLogCreationTime>2022-01-13T15:07:28+00:00</replyLog
      </stream>
    </streams>
  </netconf>
</data>
</rpc-reply>
```

NETCONF event stream

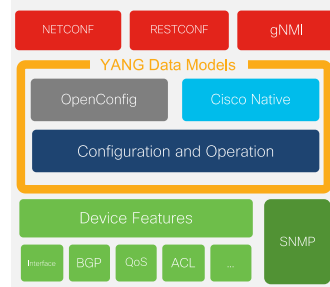
Details of the config change are sent as part of the notification

Operation = Replace
Target = Cisco-IOS-XE-Native
Xpath = /ios:native/ios:hostname

IOS XE
configuration change
CLI/API/etc



RFC5277 Event
Notification with
configuration change
details



```
jcoho-c9300#conf t
Enter configuration commands, one per line. End with CNTL/Z.
jcoho-c9300(config)#hostname jeremy_was_here
jeremy_was_here(config)#end
jeremy_was_here#
jeremy_was_here#
```

```
Sending:
#297
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:59b0d5ed-5ab9-412f-9eb1-7917db8a5c48"><create-subscription xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <stream>NETCONF/</stream>
</create-subscription>
</nc:rpc>
##
```

Received message from host

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:59b0d5ed-5ab9-412f-9eb1-7917db8a5c48">
  <ok/>
</rpc-reply>
```

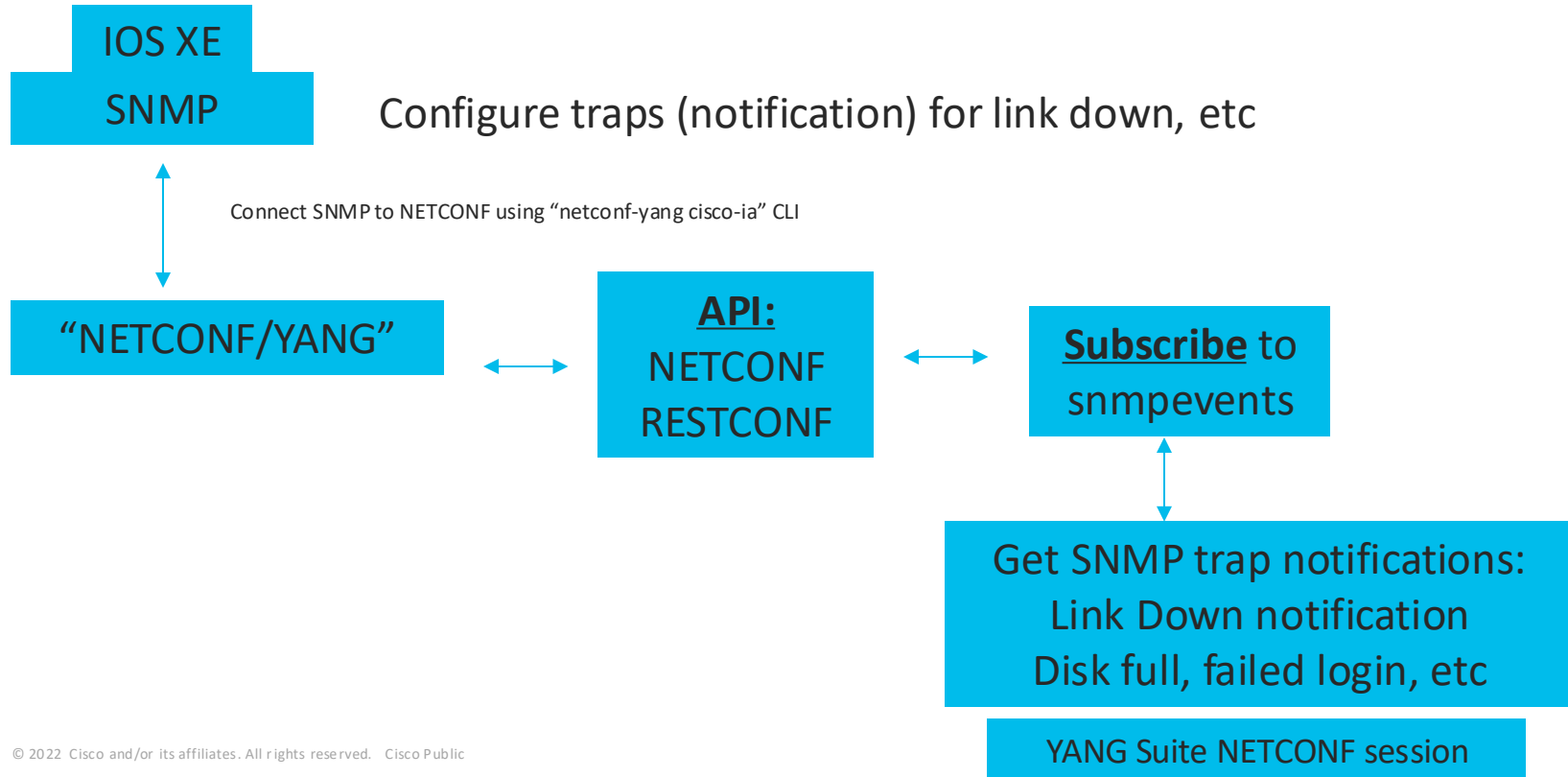
NETCONF rpc COMPLETE

Received message from host

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"><eventTime>2023-05-17T14:51:30.373451+00:00</eventTime>
<netconf-config-change xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-notifications">
  <changed-by>
    <server/>
  </changed-by>
  <datastore>running</datastore>
  <edit>
    <target xmlns:ios="http://cisco.com/ns/yang/Cisco-IOS-XE-native">/ios:native/ios:hostname</target>
    <operation>replace</operation>
  </edit>
</netconf-config-change>
</notification>
```





snmpevents events stream – block diagram



- RESTCONF

YANG Suite RESTCONF

RESTCONF provides a programmatic interface based on standard mechanisms for accessing configuration data, state data, data-model-specific Remote Procedure Call (RPC) operations and events, defined in the YANG model. The YANG Suite RESTCONF plugin provides Swagger UI and execution visualization of the YANG data model.

YANG Suite
RESTCONF




Select a YANG set:

Select a device:

Select YANG module(s):

Select depth limit:

 Cisco-IOS-XE-interfaces-oper

-  interfaces
 -  interface
 -  name
 -  interface-type

YANG Suite RESTCONF replaces the need for POSTMAN, which doesn't have integration with YANG models

Servers



default

<input type="button" value="GET"/>	/data/Cisco-IOS-XE-interfaces-oper:interfaces
<input type="button" value="GET"/>	/data/Cisco-IOS-XE-interfaces-oper:interfaces/interface
<input type="button" value="GET"/>	/data/Cisco-IOS-XE-interfaces-oper:interfaces/interface={interface-name}
<input type="button" value="GET"/>	/data/Cisco-IOS-XE-interfaces-oper:interfaces/interface={interface-name}/interface-type
<input type="button" value="GET"/>	/data/Cisco-IOS-XE-interfaces-oper:interfaces/interface={interface-name}/admin-status

Demo 1: RESTCONF

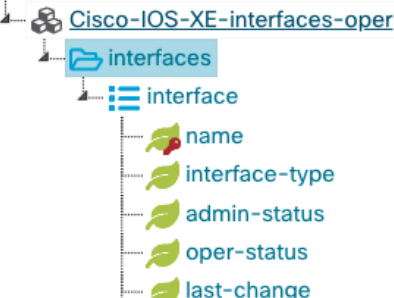
YANG Suite
RESTCONF ready

Select a YANG set:

Select a device:

Select YANG module(s):

Select depth limit:



YANG Suite
API(s) are generated

OpenAPI v3.0.3 3.0.3 OAS3

HOST DESTINATION: <https://10.1.1.5:443> (proxy through YANG Suite server)

Servers

default

- </data/Cisco-IOS-XE-interfaces-oper:interfaces>
- </data/Cisco-IOS-XE-interfaces-oper:interfaces/interface>
- </data/Cisco-IOS-XE-interfaces-oper:interfaces/interface={interface-name}>
- </data/Cisco-IOS-XE-interfaces-oper:interfaces/interface={interface-name}/interface-type>

Demo 2: Get Loopback using RESTCONF

The screenshot shows the Cisco YANG Suite RESTCONF interface. On the left is a navigation menu with the following items: Admin, Setup, Analytics, Explore, Protocols (expanded to show gNMI, gRPC telemetry, NETCONF, and RESTCONF), Test Manager, and Help. The RESTCONF section is active. The main area contains the following configuration options:

- YANG Suite RESTCONF** header with a hamburger menu icon, a globe icon, and the text "admin" with refresh, back, and help icons.
- Select a YANG set:** A dropdown menu with the value "c9300-default-yangset".
- Select a device:** A dropdown menu with the value "C9300".
- Select YANG module(s):** A dropdown menu with the value "Cisco-IOS-XE-native x".
- Select depth limit:** A dropdown menu with the value "2".
- Load module(s):** A blue button.

Demo 2: Acknowledge Notification

The screenshot shows the Cisco YANG Suite RESTCONF interface. On the left is a navigation sidebar with the following items: Admin, Setup, Analytics, Explore, Protocols (expanded to show gNMI, gRPC telemetry, NETCONF, and RESTCONF), Test Manager, and Help. The RESTCONF section is active. The main area contains the following configuration options:

- Select a YANG set: c9300-default-yangset
- Select a device: C9300
- Select YANG module(s): Cisco-IOS-XE-native x
- Select depth limit: 2

Below these options are three buttons: "Load module(s)", "Generate API(s)", and "Replays". The "Generate API(s)" button is highlighted. Below the buttons, the selected module "Cisco-IOS-XE-native" is listed. A notification box titled "YANG Suite" is displayed, containing the message: "Tree generated, please select a node to generate API(s)/script(s)."

Demo 2: Select interfaces > Loopback

Cisco YANG Suite

- Admin
- Setup
- Analytics
- Explore
- Protocols
- gNMI
- gRPC telemetry
- NETCONF
- RESTCONF
- Test Manager
- Help

Select a YANG set
c9300-default-yangset

Select a device
C9300

Select YANG module(s)
Cisco-IOS-XE-native x

Select depth limit
2

Load module(s) Generate API(s) Replays

- policy
- interface
 - AppNav-Compress
 - AppNav-UnCompress
 - ATM
 - ATM-subinterface
 - ATM-ACR
 - ATM-ACRsubinterface
 - BDI
 - CEM
 - CEM-ACR
 - Embedded-Service-Engine
 - Ethernet
 - FastEthernet
 - ucse
 - Ethernet-Internal
 - Service-Engine
 - GigabitEthernet
 - TwoGigabitEthernet
 - FiveGigabitEthernet
 - AppGigabitEthernet
 - TenGigabitEthernet
 - TwentyFiveGigabitEthernet
 - TwentyFiveGigE
 - FortyGigabitEthernet
 - FiftyGigabitEthernet
 - HundredGigE
 - TwoHundredGigE
 - FourHundredGigE
 - Loopback

Demo 2: View the Generated RESTCONF APIs

OpenAPI v3.0.3 3.0.3 OAS3

HOST DESTINATION: <https://10.1.1.5:443> (proxy through YANG Suite server)

Servers

[/restconf/proxy/https://10.1.1.5:443/restconf](https://10.1.1.5:443/restconf) - YANG SUITE Proxy RESTCONF API

default

PATCH	/data/Cisco-IOS-XE-native:native/interface/Loopback	✓	🔒
PUT	/data/Cisco-IOS-XE-native:native/interface/Loopback	✓	🔒
POST	/data/Cisco-IOS-XE-native:native/interface/Loopback	✓	🔒
GET	/data/Cisco-IOS-XE-native:native/interface/Loopback	✓	🔒
GET	/data/Cisco-IOS-XE-native:native/interface/Loopback={Loopback-name}	✓	🔒
DELETE	/data/Cisco-IOS-XE-native:native/interface/Loopback={Loopback-name}	✓	🔒
PATCH	/data/Cisco-IOS-XE-native:native/interface/Loopback={Loopback-name}/description	✓	🔒
PUT	/data/Cisco-IOS-XE-native:native/interface/Loopback={Loopback-name}/description	✓	🔒

Demo 2: RESTCONF GET Loopback API

GET /data/Cisco-IOS-XE-native:native/interface/Loopback

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Successful OK	No links
	Media type application/yang-data+json	
400	Internal error	No links
	Media type application/yang-data+json	
405	Method not allowed	No links
	Media type application/yang-data+json	
500	Internal server error	No links
	Media type application/yang-data+json	

Demo 2: Execute the RESTCONF GET Loopback API

GET /data/Cisco-IOS-XE-native:native/interface/Loopback ^

Parameters Cancel

No parameters

Execute

Responses

Demo 2: View, Copy, or Download the Response

Responses

Curl

```
curl -X 'GET' \  
'http://localhost:18480/restconf/proxy/https://10.1.1.5:443/restconf/data/Cisco-IOS-XE-native:native/interface/Loopback' \  
-H 'accept: application/yang-data+json'
```

Request URL

```
http://localhost:18480/restconf/proxy/https://10.1.1.5:443/restconf/data/Cisco-IOS-XE-native:native/interface/Loopback
```

Server response

Code **Details**

200

Response body

```
{  
  "Cisco-IOS-XE-native:Loopback": [  
    {  
      "name": 0,  
      "ip": {  
        "address": {  
          "primary": {  
            "address": "192.168.12.1",  
            "mask": "255.255.255.0"  
          }  
        }  
      },  
      "logging": {  
        "event": {  
          "link-status": [  
            null  
          ]  
        }  
      }  
    }  
  ]  
}
```

Response headers

```
cache-control: private,no-cache,must-revalidate,proxy-revalidate  
content-length: 325
```

Close

YANG Suite RESTCONF GET Interface

GET/data/Cisco-IOS-XE-interfaces-oper:interfaces/interface



Parameters

Try it out

Curl

```
curl -X 'GET' \  
  'http://localhost:8480/restconf/proxy/https://10.1.1.5:443/restconf/data/Cisco-IOS-XE-interfaces-oper:interfaces' \  
  -H 'accept: application/yang-data+json'
```

Request URL

```
http://localhost:8480/restconf/proxy/https://10.1.1.5:443/restconf/data/Cisco-IOS-XE-interfaces-oper:interfaces
```

Server response

Code	Details
------	---------

200

Response body

```
"statistics": {  
  "discontinuity-time": "2021-10-14T19:00:01+00:00",  
  "in-octets": "798",  
  "in-unicast-pkts": "9",  
  "in-broadcast-pkts": "9",  
  "in-multicast-pkts": "9",  
  "in-discards": 0,  
  "in-errors": 0,  
  "in-unknown-protos": 0,  
  "out-octets": 206032,  
  "out-unicast-pkts": "2478",
```

YANG Suite RESTCONF Demo

Cisco YANG Suite

Admin

Setup

Analytics

Explore

Protocols

Test Manager

Help

YANG Suite / Help

Help: Welcome to Cisco YANG Suite!

admin

Welcome to Cisco YANG Suite!

YANG Suite is a set of tools related to YANG models ([RFC 6020](#), [RFC 7950](#)) and related technologies such as NETCONF ([RFC 6241](#)).

It provides a modular infrastructure which various YANG application plugins can be used.

Please check [developer.cisco.com/yangsuite](#) for information, learning labs, and announcements.

Docker-based Installation

The `yangsuite/docker/start_yang_suite.sh` script performs the following:

- Prompts for username, password, and email which will be the superuser to yangsuite.
- Gives the choice of creating test SSL/TLS certificate and key.
- Creates an environment file needed for the yangsuite docker container install.
- Runs `docker-compose up`.

YANG Suite Documentation

- Welcome to Cisco YANG Suite!
- Device Profiles
 - Managing device profiles
 - Setting up YANG Suite TLS/SSL
- File Manager
 - Constructing and populating a YANG module repository
 - Uploading YANG files from the local filesystem to a YANG repository
 - Downloading YANG files via NETCONF from a device to a YANG repository
 - Copying YANG modules via SCP from a server to a YANG repository
 - Importing YANG modules from a Git repository to a YANG repository
 - Defining a YANG module set
 - Managing YANG module files in YANG Suite
- Working with YANG Models
 - Exploring YANG Models
- YANG Suite Analytics
 - Datasets
- Using gNMI with YANG Suite
- YANG Suite gRPC telemetry receivers
 - Telemetry over gRPC Clear Channel
 - Configuring gRPC Telemetry Receivers
 - Telemetry over gRPC Secure Channel
- Using NETCONF with YANG Suite
 - Using NETCONF RPCs
 - Working with NETCONF Notification Streams
 - Using Replays for repeated workflows
 - Locking and unlocking datastores
- RESTCONF in YANG Suite
 - YANG Suite RESTCONF
- Test management with YANG Suite
 - Using Test Manager to define and execute tests
 - Convert Ytool Test Suites to YANG Suite Formats to activate Windows.
 - Generating YANG model Tests
 - Importing and exporting YANG Suite tests

RESTCONF + Python Script



Select a YANG set

c9300-default-yangset

Select YANG module(s)

Cisco-IOS-XE-interfaces-oper x

Select a device

C9300

Select depth limit

No limit

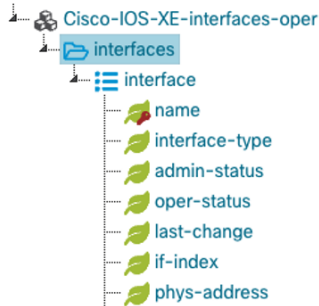
Load module(s)

Generate API(s)

Show API(s)

Generate Python Script

Download Ansible Playbook



RESTCONF + Ansible

YANG Suite
RESTCONF

Select a YANG set: jcohoe-c9300-default-yangset

Select a device: jcohoe-c9840

Select YANG module(s): Cisco-IOS-XE-interfaces-oper

Select depth limit: No limit

Buttons: Load module(s), Generate API(s), Show API(s), Generate Python Script, Download Ansible Playbook

Tree view: Cisco-IOS-XE-interfaces-oper > interfaces > interface > name, interface-type, admin-status

Requirements

1. Install Ansible
2. Install the required collection:
ansible-galaxy
collection
install
ansible.netcommon

Download Ansible Playbook

Fill Out All Fields

Selected XPath: /interfaces

Script File Name: script_name

Ansi Task Name: interfaces_oper

REST Message Name: interface

Method: GET

Buttons: Download, Close

```
- name: interface
hosts: HOST_NAME_HERE
gather_facts: no
tasks:
- name: interfaces_oper
ansible.netcommon.restconf_get:
# Output can either be json or xml
output: json
path: Cisco-IOS-XE-interfaces-oper:interfaces
```

RESTCONF + Ansible

The screenshot displays the Cisco YANG Suite RESTCONF interface. On the left is a navigation menu with options: Admin, Setup, Analytics, Explore, Protocols, gNMI, gRPC telemetry, NETCONF, and RESTCONF. The main header shows 'YANG Suite RESTCONF' with a 'ready' status and an 'admin' user profile. The configuration area includes four dropdown menus: 'Select a YANG set' (c9300-default-yangset), 'Select a device' (C9300), 'Select YANG module(s)' (Cisco-IOS-XE-interfaces), and 'Select depth limit' (No limit). Below this, a 'default' section shows a PATCH request configuration for the path '/data/Cisco-IOS-XE-native:hostname'. The 'Request body' is set to 'application/yang-data+json' and contains the JSON: { "hostname": "restconf-test" }. The interface includes 'Cancel' and 'Reset' buttons for parameters, and 'Execute' and 'Clear' buttons at the bottom.

```
PATCH /data/Cisco-IOS-XE-native:hostname
```

Parameters Cancel Reset

No parameters

Request body required application/yang-data+json

Set system's network name

```
{
  "hostname": "restconf-test"
}
```

Execute Clear

```
Result
C9300#
C9300#
restconf-test#
restconf-test#
```

- gNMI

SSL/TLS for gNMI

When creating or edit device details, the TLS Authority Cert, TLS Client Cert, and TLS Client Key are used for the gNMI session

gNMI +
gRPC Dial-Out

TLS Authority Certificate:

No file chosen

TLS Client Certificate:

No file chosen

TLS Client Key:

No file chosen

gRPC Dial-Out

TLS Server Certificate:

No file chosen

TLS Server Key:

No file chosen

YANG Suite gNMI

Model-driven configuration and retrieval of config and operational data using the gRPC Network Management Interface (gNMI) Capabilities, Get, Set and Subscribe remote procedure calls (RPCs).
This fully functional gNMI client helps build, test, and validate gNMI YANG payloads

The screenshot displays the YANG Suite gNMI interface. At the top, the breadcrumb is "YANG Suite / gNMI / YANG set "" / Modules". The main title is "gRPC Network Management Interface (gNMI)" with a user "admin" in the top right. Below the title, there are dropdowns for "YANG Set" (c9300-default-yangset) and "Module(s)" (Cisco-IOS-XE-interfaces-oper). The "Device" is set to "C9300". There are buttons for "Edit Device" and "Capabilities". The "gNMI Operation" section has buttons for "Get", "Set", and "Subscribe", with "All" selected. The "Origin" section has radio buttons for "Openconfig", "RFC 7951" (selected), and "Other". The "Encoding type" section has radio buttons for "JSON_IETF" (selected), "JSON", and "Prefixing". There are buttons for "Search XPath...", "Show Legend", "Build JSON", "Clear Values", "Run RPC(s)", and "Clear RPC(s)".

Nodes	Value	Operation
Cisco-IOS-XE-interfaces-oper		
interfaces	<input checked="" type="checkbox"/>	
interface	<input checked="" type="checkbox"/>	
name	<input type="text" value="*"/>	
interface-type		
admin-status		
oper-status		
last-change		
if-index		

```
gNMI GET Response
=====
notification {
  timestamp: 1634241613977826499
  update {
    path {
      origin: "rfc7951"
      elem {
        name: "Cisco-IOS-XE-interfaces-oper:interfaces"
      }
      elem {
        name: "interface"
        key {
          key: "name"
          value: "AppGigabitEthernet1/0/1"
        }
      }
    }
  }
}
val {
  json_ietf_val: "{\"name\": \"AppGigabitEthernet1/0/1\", \"
```

Demo: gNMI GET for Cisco Native hostname

YANG Suite / gNMI / YANG set "" / Modules

admin

gRPC Network Management Interface (gNMI)

YANG Set: c9300-default-yangset Module(s): Cisco-IOS-XE-interfaces-oper Load Module(s)

Device: C9300 Edit Device Capabilities

gNMI Operation: Get Set Subscribe All Config State Operational

Origin: Openconfig RFC 7951 Other Encoding type: JSON_IETF JSON Prefixing

Search XPath(s)... Show Legend Run RPC(s)

Build JSON Clear Values

Nodes	Value	
Cisco-IOS-XE-interfaces-oper		path { origin: "rfc7951" elem { name: "interfaces" } }
interfaces	<input checked="" type="checkbox"/>	encoding: JSON_IETF
interface		
name		
interface-type		

Stop Session Polling count: 65 Received bytes of data: 681356

```
gNMI GET
=====
path {
  origin: "rfc7951"
  elem {
    name: "interfaces"
  }
}
encoding: JSON_IETF

gNMI GET Response
=====
notification {
  timestamp: 1637001179376600080
  update {
    path {
      origin: "rfc7951"
      elem {
        name: "Cisco-IOS-XE-interfaces-oper:interfaces"
      }
    }
    val {
      json_ietf_val: "{\ninterface\": [{\nname\": \"AppGigabitEthernet1/0/1\", \ninterface-type\": \"iana-iftype\"}]"
    }
  }
  update {
    path {
```

Demo: gNMI SET for Cisco Native hostname

The screenshot displays the Cisco YANG Suite gNMI Network Management Interface (gNMI) for configuring a Cisco IOS-XE native hostname. The interface includes a sidebar with navigation options (Admin, Setup, Analytics, Explore, Protocols, gNMI, gRPC telemetry, NETCONF, RESTCONF, Test Manager, Help) and a main configuration area.

Configuration Details:

- YANG Set:** jchohoe-c9300-default-yangset
- Module(s):** Cisco-IOS-XE-native
- Device:** jchohoe-c9300
- Origin:** RFC 7951 (selected)
- Encoding type:** JSON_IETF (selected)
- Buttons:** Edit Device, Capabilities, gNMI Operation (Get, Set, Subscribe), Prefix path, Base64, Run RPC(s), Clear RPC(s), Clear Values, Build RPC.

Configuration Tree:

- Nodes
 - parser
 - service
 - platform
 - hostname (selected)
 - enable
 - password
 - eap
 - archive
 - username
 - user-name
 - card
 - controller
 - frame-relay
 - aqm-register-fnf
 - vrf
 - global-address-family
 - rmon
 - sampler
 - flow

Configuration Value: set-by-ys-gnmi

RPC Payload (JSON):

```
{
  "update": [
    {
      "path": {
        "origin": "rfc7951",
        "elem": [
          {
            "name": "Cisco-IOS-XE-native:native"
          }
        ]
      },
      "val": {
        "jsonIetfVal": {
          "hostname": "set-by-ys-gnmi"
        }
      }
    }
  ]
}
```

Demo: gNMI GET OC Hostname

The screenshot shows the gRPC Network Management Interface (gNMI) for the 'openconfig-system' module on device 'jcohoe-c9300'. The 'GET type' is set to 'Openconfig' and the 'Encoding type' is 'JSON_IETF'. The 'hostname' node under 'config' is selected, with the value 'oc-inet:domain-name' displayed. The 'Run RPC(s)' button is highlighted in blue.

YANG Suite / gNMI / YANG set "" / Modules
gRPC Network Management Interface (gNMI) admin

YANG Set: jcohoe-c9300-default-yangset Module(s): openconfig-system Load Module(s)

Device: jcohoe-c9300 Edit Device Capabilities gNMI Operation Get Set Subscribe Prefix path

GET type: All Config State Operational Openconfig RFC 7951 Module name Other Encoding type: JSON_IETF JSON PROTO ASCII

Search XPath(s) Legend Replays Build RPC Clear Values Run RPC(s) Clear RPC(s)

Nodes: Value Operation

- openconfig-system
 - system
 - config
 - hostname: oc-inet:domain-name
 - domain-name
 - login-banner
 - motd-banner
 - state
 - clock
 - dns
 - ntp
 - grpc-server

```
{ "path": [ { "origin": "openconfig", "elem": [ { "name": "system" }, { "name": "config" }, { "name": "hostname" } ] }, ], "encoding": "JSON_IETF" }
```

```
gNMI GET Response
=====
notification {
  timestamp: 1663716002850520766
  update {
    path {
      origin: "openconfig"
      elem {
        name: "system"
      }
      elem {
        name: "config"
      }
      elem {
        name: "hostname"
      }
    }
    val {
      json_ietf_val: "\"set-by-ys-gnmi-ansible\""
    }
  }
}
```

```
JSON Decoded
=====
"set-by-ys-gnmi-ansible"
```

gNMI GET IETF Interfaces

The screenshot displays the Cisco YANG Suite interface for gNMI. The left sidebar shows the navigation menu with 'gNMI' selected. The main area is titled 'gRPC Network Management Interface (gNMI)'. The configuration includes:

- YANG Set:** c9300-default-yangset
- Module(s):** ietf-interfaces
- Device:** C9300
- gNMI Operation:** Get, Set, Subscribe
- GET type:** All (selected), Config, State, Operational, Openconfig, RFC 7951, Module name, Other
- Origin:** RFC 7951 (selected)
- Encoding type:** JSON_IETF (selected), JSON, PROTO

Buttons for 'Search XPath(s)', 'Legend', 'Replays', 'Build RPC', 'Clear Values', 'Run RPC(s)', and 'Clear RPC(s)' are visible. The tree view on the left shows the 'ietf-interfaces' module expanded to 'interfaces', with 'interface' selected. The right pane displays the resulting JSON RPC response:

```
{
  "path": [
    {
      "origin": "rfc7951",
      "elem": [
        {
          "name": "ietf-interfaces:interfaces"
        }
      ]
    }
  ],
  "encoding": "JSON_IETF"
}
```

gNMI GET IETF Interfaces Response

Stop Session

Session status: running

Received bytes of data: 14432

Connecting insecure channel

```
gNMI GET
=====
path {
  origin: "rfc7951"
  elem {
    name: "ietf-interfaces:interfaces"
  }
}
encoding: JSON_IETF
```




```
gNMI GET Response
=====
notification {
  timestamp: 1661543913969767107
  update {
    path {
      origin: "rfc7951"
      elem {
        name: "ietf-interfaces:interfaces"
      }
    }
    val {
      json_ietf_val: "{\"interface\": [{\"name\": \"FortyGigabitEthernet1/1/1\", \"type\": \"iana-if-type:ethernetCsmacd\", \"enabled\": true, \"ietf-ip:ipv4\": {}, \"ietf-if-\"}]}\"
    }
  }
}
```

JSON Decoded

```
=====
{
  "interface": [
    {
      "name": "FortyGigabitEthernet1/1/1",
      "type": "iana-if-type:ethernetCsmacd",
      "enabled": true,
      "ietf-ip:ipv4": {},
      "ietf-ip:ipv6": {}
    },
    {
      "name": "FortyGigabitEthernet1/1/2",
      "type": "iana-if-type:ethernetCsmacd",
      "enabled": true,
      "ietf-ip:ipv4": {},
      "ietf-ip:ipv6": {}
    }
  ],
}
```

Demo: gNMI SET OC Hostname

YANG Suite / gNMI / YANG set "" / Modules

admin   

YANG Set: Module(s): Load Module(s)

Device: Edit Device Capabilities gNMI Operation Get Set Subscribe Prefix path Base64

Origin: Openconfig RFC 7951 Module name Other Encoding type: JSON_IETF JSON PROTO ASCII

Search XPath(s) Legend Replays Build RPC Clear Values Run RPC(s) Clear RPC(s)

Nodes	Value	Operation
openconfig-system		
system		
config		
hostname	<input type="text" value="set-by-ys-gnmi"/>	<input type="text" value="update"/>
domain-name		
login-banner		
motd-banner		
state		
clock		
dns		
ntp		
grpc-server		
ssh-server		
telnet-server		
logging		

```
{
  "update": [
    {
      "path": {
        "origin": "openconfig",
        "elem": [
          {
            "name": "system"
          },
          {
            "name": "config"
          }
        ]
      },
      "val": {
        "jsonIetfVal": {
          "hostname": "set-by-ys-gnmi-ansible"
        }
      }
    }
  ]
}
```


gNMI Subscribe OpenConfig interfaces

YANG Suite / gNMI / YANG set ** / Modules
gRPC Network Management Interface (gNMI) admin

YANG Set: jcohoec-c9300-default-yangset Module(s): openconfig-interfaces Load Module(s)

Device: jcohoec-c9300 Edit Device Capabilities gNMI Operation: Get Set Subscribe Prefix path

STREAM ONCE POLL
ON_CHANGE SAMPLE TARGET_DEFINED Sample interval: 30 Origin: Openconfig RFC 7951 Module name Other

Encoding type: JSON_IETF JSON PROTO ASCII

Search XPath(s) Legend Replays Build RPC Clear Values Run RPC(s) Clear RPC(s)

Nodes	Value	Operation
openconfig-interfaces		
interfaces		
interface		
name	✓	
config	✓	
state		
subinterfaces		
oc-eth:ethernet		
oc-lag:aggregation		
oc-vlan:routed-vlan		
hold-time		

```
{
  "subscribe": {
    "subscription": [
      {
        "path": {
          "origin": "openconfig",
          "elem": [
            {
              "name": "interfaces"
            },
            {
              "name": "interface"
            }
          ]
        },
        "mode": "SAMPLE",
        "sampleinterval": "30000000000"
      }
    ],
    "encoding": "JSON_IETF"
  }
}
```

```
update {
  timestamp: 1663716380431805000
  update {
    path {
      origin: "openconfig"
      elem {
        name: "interfaces"
      }
      elem {
        name: "interface"
      }
    }
  }
}

JSON Decoded
=====
val [
] = {
  {
    "name": "FortyGigabitEthernet1/1/1",
    "config": {
      "name": "FortyGigabitEthernet1/1/1",
      "type": "iana-if-type:ethernetCsmacd",
      "enabled": true
    },
    "state": {
      "name": "FortyGigabitEthernet1/1/1",
      "type": "iana-if-type:ethernetCsmacd",
      "enabled": true,
      "ifindex": 45,
      "admin-status": "UP",
      "oper-status": "NOT_PRESENT",
      "last-change": "1658857381366000000",
      "counters": {
        "in-octets": "0",
        "in-unicast-pkts": "0",
        "in-broadcast-pkts": "0",
        "in-multicast-pkts": "0",
        "in-discards": "0",
        "in-errors": "0",
        "in-unknown-protos": "0",
        "in-fcs-errors": "0",
        "out-octets": "0",
        "out-unicast-pkts": "0",
        "out-broadcast-pkts": "0",
        "out-multicast-pkts": "0",
        "out-discards": "0",
        "out-errors": "0",
        "last-clear": "1658857257000000000"
      }
    }
  },
  "openconfig-platform-port:hardware-port": "FortyGigabitEthernet1/1/1"
},
}
```

gNMI + Ansible

Requirements

1. Install Ansible
2. Install the required collection

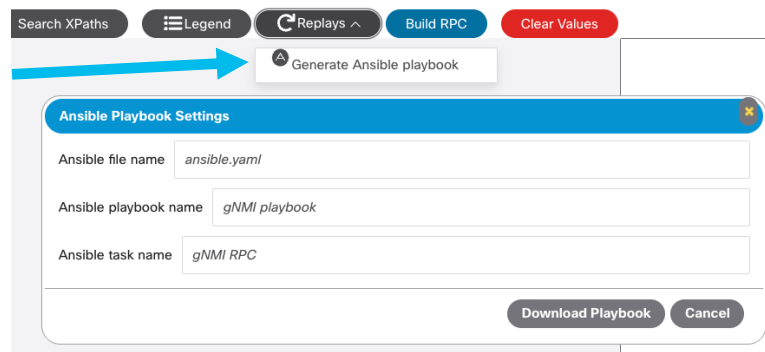
```
ansible-galaxy collection install nokia.grpc
```

```
- name : gNMI playbook
  gather_facts: false
  hosts: MY_HOST_NAME

collections:
- nokia.grpc

tasks:
- name: gNMI RPC
  gnmi_config:
    update:
      - path: system/config/hostname
        val: set-by-ys-gnmi-ansible

  register: testout
- name: dump test output
  debug:
    msg: '{{ testout.output }}'
```



```
auto@pod19-xelab:~/ansible/YANGSuite-ansible$ ansible-galaxy collection install nokia.grpc
Process install dependency map
Starting collection install process
Installing 'nokia.grpc:1.0.2' to '/home/auto/.ansible/collections/ansible_collections/nokia/grpc'
auto@pod19-xelab:~/ansible/YANGSuite-ansible$
```

Note: the ansible gnmi integration works only with OpenConfig model

gNMI + Ansible demo

The screenshot displays the Cisco YANG Suite gRPC Network Management Interface (gNMI) for a specific YANG Set. The interface includes a navigation sidebar on the left with categories like Admin, Setup, Analytics, Explore, Protocols, Test Manager, and Help. The main content area shows the following configuration details:

- YANG Set:** 1711-c9300
- Module(s):** openconfig-system
- Device:** (none selected)
- Capabilities:** gNMI Operation
- GET type:** All (selected), Config, State, Operational, Openconfig, RFC 7951, Module name, Other
- Origin:** (empty)
- Encoding type:** JSON_IETF (selected), JSON, PROTO, ASCII
- Buttons:** Search XPath, Legend, Replays, Build RPC, Clear Values, Run RPC(s), Clear RPC(s), Load Module(s)
- Nodes:** openconfig-system

The interface is designed for managing network configurations via gRPC, providing various options for data retrieval and manipulation.

- gRPC Telemetry

gRPC Dial-Out with TLS Support

- The gRPC Telemetry receiver is accessed through Protocols tab and supports multiple gRPC telemetry receivers
- The Listen IP and port must be specified and TLS certifications can optionally be used to secure the telemetry data in transit

The screenshot shows the Cisco YANG Suite interface for configuring gRPC Telemetry. On the left is a navigation menu with 'gRPC telemetry' highlighted. The main area shows configuration fields for 'Listen on IP address' (0.0.0.0) and 'Listen at port' (57500). There are buttons for 'Start receiver', 'Manage receivers', and 'Clear'. Below this is a 'Telemetry Receivers' dialog box with fields for 'Output file', 'Elasticsearch Output URI', and a table for 'IP Address Port TLS' with values 0.0.0.0, 57500, false, and Stop.

Cisco YANG Suite

- Admin
- Setup
- Analytics
 - Datasets and diffs
 - SNMP to YANG Mapping
 - YANG coverage
- Explore
 - YANG
- Protocols
 - gNMI
 - gRPC telemetry**
 - NETCONF
 - RESTCONF
- Test Manager
- Help

YANG Suite / gRPC Telemetry
gRPC Telemetry

admin

Listen on IP address: 0.0.0.0 | Listen at port: 57500

Start receiver (Optional) TLS receiver Manage receivers

Clear

Telemetry Receivers

Output file
File path for telemetry output.

Elasticsearch Output URI
example: http://localhost:9200

Set Output(s)

IP Address	Port	TLS
0.0.0.0	57500	false

Stop

Exit

gRPC Dial-Out with TLS Support

Cisco YANG Suite

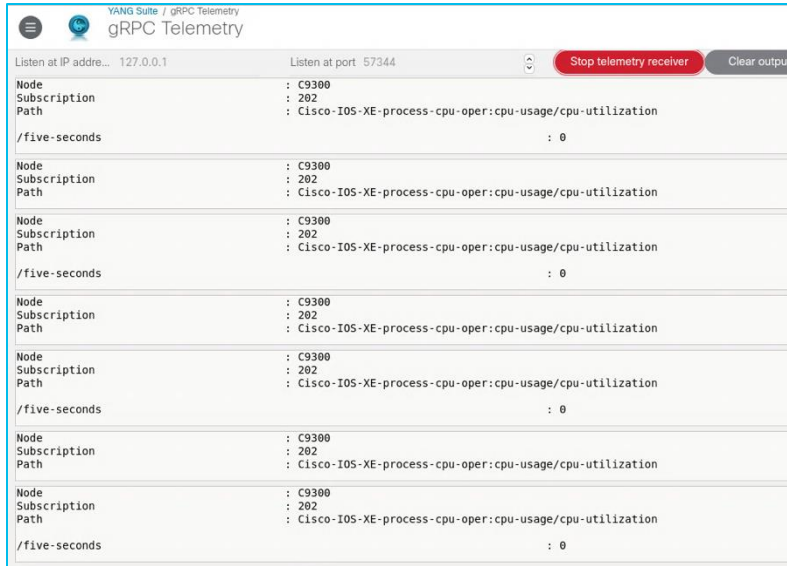
- Admin
- Setup
- Analytics
- Datasets and diffs
- SNMP to YANG Mapping
- YANG coverage
- Explore
 - YANG
- Protocols
 - gNMI
 - gRPC telemetry**
 - NETCONF
 - RESTCONF
- Test Manager
- Help

Server Certificate and Key can now be provided within the Device Profile
These certificates are used to secure the model driven telemetry data between YANG Suite and IOS XE

The screenshot displays the 'gRPC Telemetry' configuration page in the Cisco YANG Suite. The interface is divided into several sections:

- Top Bar:** 'YANG Suite / gRPC Telemetry' and 'yangsuite-developer'.
- Listen Settings:** 'Listen at IP address' (with 'IP address' input) and 'Listen at port' (with 'TCP port' input). A 'Start receiver' button is present, along with an '(Optional) TLS receiver' checkbox (highlighted with an orange box) and 'Show receivers' and 'Clear' buttons.
- TLS Configuration:** Fields for 'TLS Root Certificate: rootCA.pem', 'TLS Client Certificate: client.crt', 'TLS Client Key: client.key', 'TLS Server Certificate: server.cert', and 'TLS Server Key: server.key', each with a 'Choose File' button and 'No file chosen' status.
- Telemetry Receivers:** A table with columns 'IP Address', 'Port', and 'TLS'. It shows two receivers: one at 10.19.198.133:50070 (TLS: true) and another at 10.19.198.133:57344 (TLS: false). A 'Set Output' button and 'Exit' button are also visible.
- Device Selection:** A 'Choose device with certificate/key' dropdown menu showing 'ddmi-9500-2' selected, with a 'Start TLS telemetry receiver' button below it.
- Data Output:** A 'clear.log' window showing received JSON data. The data includes fields like 'timestamp', 'subscription', 'node', 'path', and 'value', representing telemetry data for 'Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization'.

Demo: Model Driven Telemetry - gRPC



The screenshot shows the 'gRPC Telemetry' interface with a table of subscriptions. The table has columns for Node, Subscription, Path, and /five-seconds. All entries show Node: C9300, Subscription: 202, Path: Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization, and /five-seconds: 0.

Node	Subscription	Path	/five-seconds
C9300	202	Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization	0
C9300	202	Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization	0
C9300	202	Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization	0
C9300	202	Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization	0
C9300	202	Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization	0
C9300	202	Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization	0
C9300	202	Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization	0

```
telemetry ietf subscription 202
  encoding encode-kvgpb
  filter xpath /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds
  stream yang-push
  update-policy periodic 1000
  receiver ip address 10.1.1.3 57344 protocol grpc-tcp
```

<https://github.com/jeremycohoe/cisco-ios-xe-mdt>

```
C9300#conf t
Enter configuration commands, one per line. End with CNTL/Z.
C9300(config)#telemetry ietf subscription 202
C9300(config-mdt-subs)# encoding encode-kvgpb
C9300(config-mdt-subs)# filter xpath /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds
C9300(config-mdt-subs)# stream yang-push
C9300(config-mdt-subs)# update-policy periodic 1000
C9300(config-mdt-subs)# receiver ip address 10.1.1.3 57344 protocol grpc-tcp
C9300#show telemetry ietf subscription 202 detail
Telemetry subscription detail:

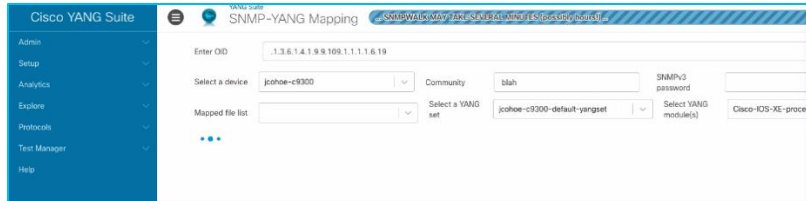
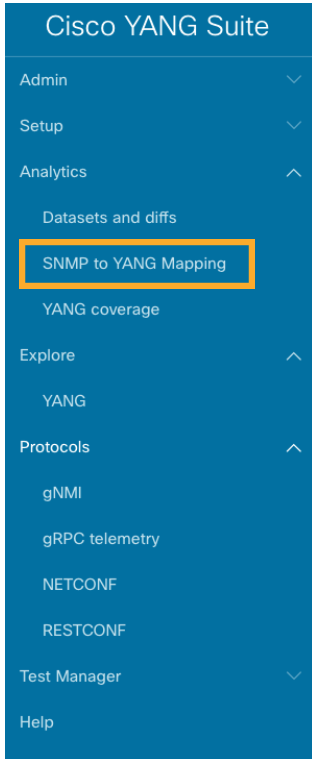
Subscription ID: 202
Type: Configured
State: Valid
Stream: yang-push
Filter:
  Filter type: xpath
  XPath: /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds
Update policy:
  Update Trigger: periodic
  Period: 1000
Encoding: encode-kvgpb
Source VRF:
Source Address:
Notes:

Legacy Receivers:
  Address          Port    Protocol  Protocol Profile
-----
10.1.1.3          57344  grpc-tcp
```

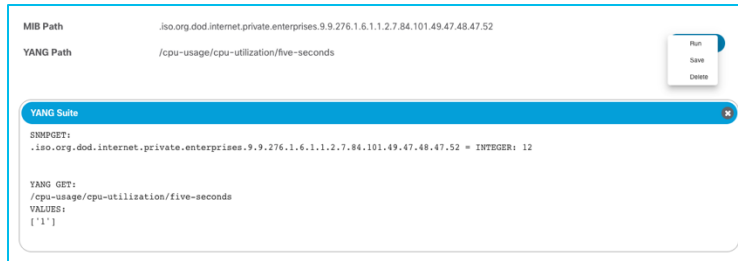
- SNMP OID to YANG XPATH

SNMP to YANG migration mapping

Ease the transition from SNMP OID to YANG Xpath and easily verify the responses from both.



CPU OID: .1.3.6.1.4.1.9.9.109.1.1.1.1.6.19
Environment OID: .1.3.6.1.4.1.9.9.91.1.1.1.1.4



Right click > Run to retrieve from SNMP and NETCONF simultaneously.

This solution utilizes the Python library for “fuzzy matching” of OID and XPATH values to identify most accurate match.

SNMP OID's mapped to YANG Xpaths

- Over 200 SNMP OIDs have been mapped to YANG Xpaths to make the SNMP to YANG transition easier
- Using YANG instead of SNMP saves customers time and money because more data is sent at faster rates using YANG API's compared to SNMP
- YANG is easier to understand and work with because Xpath format describes the feature, while SNMP requires additional mapping and prior knowledge of MIBs and OID's

- | | |
|----------------|--------------------|
| 1. BFD | 12. Memory |
| 2. CDP | 13. NVE |
| 3. CFM | 14. OSPF |
| 4. EFP | 15. OpenConfig BFD |
| 5. ENV | 16. Process CPU |
| 6. Environment | 17. Process Memory |
| 7. HSRP | 18. PoE |
| 8. Interface | 19. PTP |
| 9. L2VPN | 20. PTP Synce |
| 10. LISP | 21. Tunnel |
| 11. LLDP | |

#	Model	Example OID	# OIDs Mapped
1	Cisco-IOS-XE-bfd-oper	.1.3.6.1.4.1.9.10.137.1.3.1.1	5
2	Cisco-IOS-XE-cdp-oper	.1.3.6.1.4.1.9.9.23.1.2.1.1.4	9
3	Cisco-IOS-XE-cfm-oper	.1.3.111.2.802.1.1.8.1.7.1.1.5	11
4	Cisco-IOS-XE-efp-oper	.1.3.6.1.4.1.9.9.613.1.4.1.1.6	1
5	Cisco-IOS-XE-environment-oper	.1.3.6.1.2.1.97.1.5.3.1.1.	1
6	Cisco-IOS-XE-hsrp-oper	.1.3.6.1.4.1.9.9.106.1.2.1.1.13	6
7	Cisco-IOS-XE-interfaces-oper	.1.3.6.1.2.1.31.1.1.1.19	10
8	Cisco-IOS-XE-l2vpn-oper	.1.3.6.1.4.1.9.10.106.1.2.1.12	15
9	Cisco-IOS-XE-lisp-oper	.1.3.6.1.2.1.220.1.8.1.4.	46
10	Cisco-IOS-XE-lldp & oper	.1.0.8802.1.1.2.1.1.1	15
11	Cisco-IOS-XE-memory-oper	.1.3.6.1.4.1.9.9.48.1.1.1.6	4
12	Cisco-IOS-XE-nve-oper	.1.3.6.1.4.1.9.9.820.1.1.1.1.2	19
13	Cisco-IOS-XE-ospf-oper	.1.3.6.1.2.1.47.1.3.3.1.1	9
14	Cisco-IOS-XE-poe-oper	.1.3.6.1.2.1.105.1.3.1.1.4.	2
15	Cisco-IOS-XE-process-cpu-oper	.1.3.6.1.4.1.9.9.109.1.1.1.1.7	11
16	Cisco-IOS-XE-process-memory-oper	.1.3.6.1.4.1.9.9.109.1.2.3.1.18	5
17	Cisco-IOS-XE-ntp-synce-oper	.1.3.6.1.4.1.9.9.760.1.2.1.1.4	12
18	Cisco-IOS-XE-tunnel-oper	.1.3.6.1.2.1.10.131.1.1.1.1.3	3
19	openconfig-bfd	.1.3.6.1.4.1.9.10.137.1.3.1.6	7

SNMP OID to YANG XPATH mapping

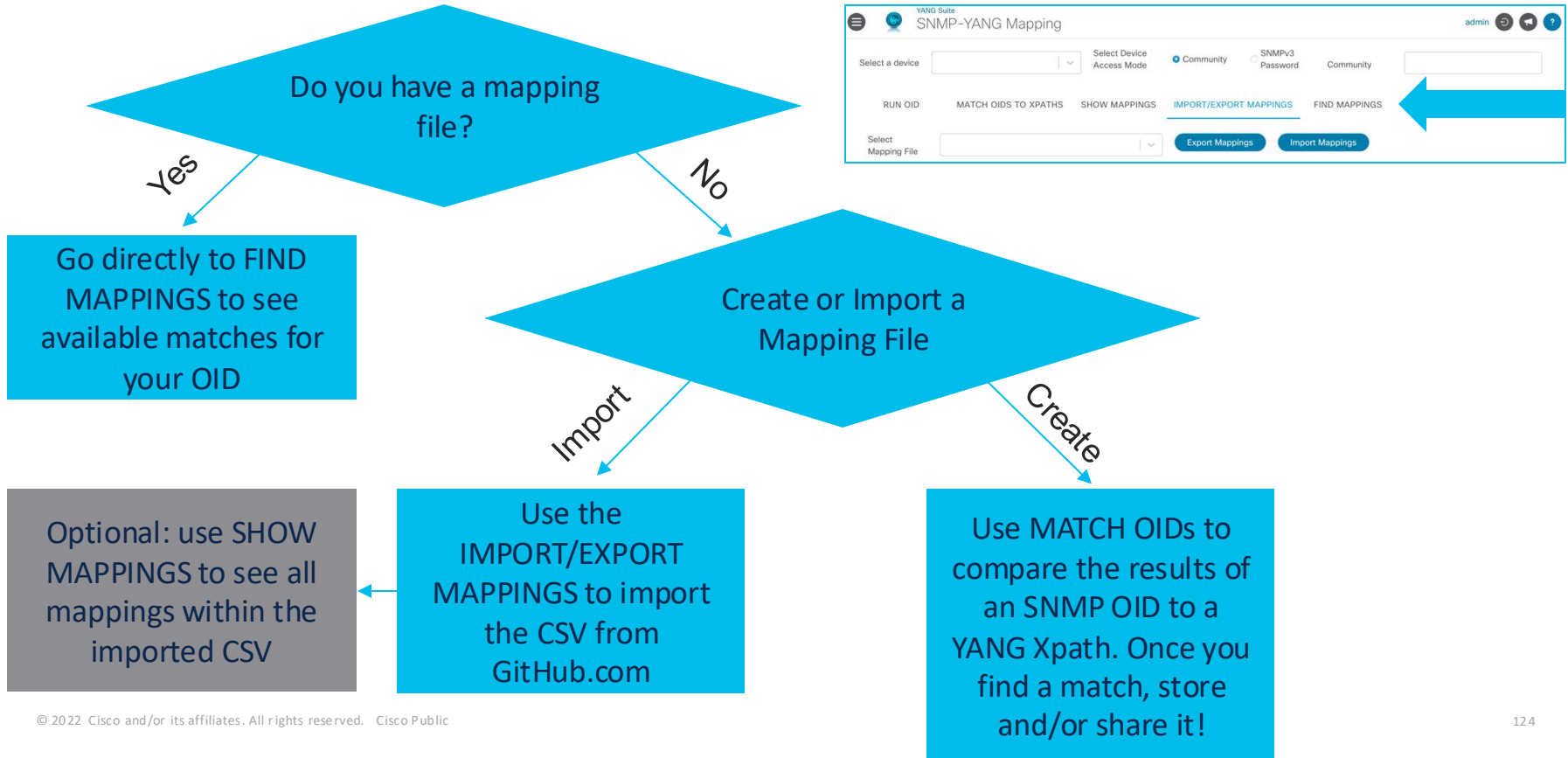
#	Model	Example OID	# OIDs Mapped
1	Cisco-IOS-XE-bfd-oper	.1.3.6.1.4.1.9.10.137.1.3.1.1	5
2	Cisco-IOS-XE-cdp-oper	.1.3.6.1.4.1.9.9.23.1.2.1.1.4	9
3	Cisco-IOS-XE-cfm-oper	.1.3.111.2.802.1.1.8.1.7.1.1.5	11
4	Cisco-IOS-XE-efp-oper	.1.3.6.1.4.1.9.9.613.1.4.1.1.6	1
5	Cisco-IOS-XE-environment-oper	.1.3.6.1.2.1.97.1.5.3.1.1	1
6	Cisco-IOS-XE-hsrp-oper	.1.3.6.1.4.1.9.9.106.1.2.1.1.13	6
7	Cisco-IOS-XE-interfaces-oper	.1.3.6.1.2.1.31.1.1.1.19	10
8	Cisco-IOS-XE-l2vpn-oper	.1.3.6.1.4.1.9.10.106.1.2.1.12	15
9	Cisco-IOS-XE-lacp-oper	.1.2.840.10006.300.43.1.2.1.1.5	10
10	Cisco-IOS-XE-lsp-oper	.1.3.6.1.2.1.220.1.8.1.4	46
11	Cisco-IOS-XE-ldp & oper	.1.0.8802.1.1.2.1.1.1	15
12	Cisco-IOS-XE-memory-oper	.1.3.6.1.4.1.9.9.48.1.1.1.6	4
13	Cisco-IOS-XE-nve-oper	.1.3.6.1.4.1.9.9.820.1.1.1.1.2	19
14	Cisco-IOS-XE-ntp-oper	1.3.6.1.2.1.197.1.2.5	10
15	Cisco-IOS-XE-ospf-oper	.1.3.6.1.2.1.47.1.3.3.1.1	9
16	Cisco-IOS-XE-poe-oper	.1.3.6.1.2.1.105.1.3.1.1.4	2
17	Cisco-IOS-XE-process-cpu-oper	.1.3.6.1.4.1.9.9.109.1.1.1.1.7	11
18	Cisco-IOS-XE-process-memory-oper	.1.3.6.1.4.1.9.9.109.1.2.3.1.18	5
19	Cisco-IOS-XE-ptp-synce-oper	.1.3.6.1.4.1.9.9.760.1.2.1.1.4	12
20	Cisco-IOS-XE-tunnel-oper	.1.3.6.1.2.1.10.131.1.1.1.1.3	3
21	openconfig-bfd	.1.3.6.1.4.1.9.10.137.1.3.1.6	1

- 20 OID's from the **LACP** and **NTP** features have been mapped
- A total of **21 feature YANG models** have been mapped to SNMP OID's
- Total number of OID's mapped = **213**

Numeric OID	YANG Xpath	YANG Module
.1.2.840.10006.300.43.1.2.1.1.5	/lag-oper-data/lacp-port-channel(key=value)/lacp-member-state(key=value)/oper-key	Cisco-IOS-XE-lacp-oper
.1.2.840.10006.300.43.1.2.1.1.3	/lag-oper-data/lacp-port-channel(key=value)/lacp-member-state(key=value)/system-id	Cisco-IOS-XE-lacp-oper
.1.2.840.10006.300.43.1.2.1.1.9	/lag-oper-data/lacp-port-channel(key=value)/lacp-member-state(key=value)/partner-id	Cisco-IOS-XE-lacp-oper
.1.2.840.10006.300.43.1.2.1.1.14	/lag-oper-data/lacp-port-channel(key=value)/lacp-member-state(key=value)/port-num	Cisco-IOS-XE-lacp-oper
.1.2.840.10006.300.43.1.2.1.1.17	/lag-oper-data/lacp-port-channel(key=value)/lacp-member-state(key=value)/partner-port-num	Cisco-IOS-XE-lacp-oper
.1.2.840.10006.300.43.1.1.1.1.6	/lag-oper-data/lag-info(key=value)/channel-group	Cisco-IOS-XE-lacp-oper
.1.2.840.10006.300.43.1.1.1.1.5	/lag-oper-data/lag-info(key=value)/port-channel-up	Cisco-IOS-XE-lacp-oper
.1.2.840.10006.300.43.1.2.2.1.1	/lag-oper-data/lacp-port-channel(key=value)/lacp-member-state(key=value)/counters/lacp-in-pkts	Cisco-IOS-XE-lacp-oper
.1.2.840.10006.300.43.1.2.2.1.6	/lag-oper-data/lacp-port-channel(key=value)/lacp-member-state(key=value)/counters/lacp-out-pkts	Cisco-IOS-XE-lacp-oper
.1.2.840.10006.300.43.1.2.1.1.11	/lag-oper-data/lacp-port-channel(key=value)/lacp-member-state(key=value)/partner-key	Cisco-IOS-XE-lacp-oper

Numeric OID	YANG Xpath	YANG Module
1.3.6.1.2.1.197.1.2.4	/ntp-oper-data/ntp-status-info/refid/ip-addr	Cisco-IOS-XE-ntp-oper
1.3.6.1.2.1.197.1.2.9	/ntp-oper-data/ntp-status-info/reftime	Cisco-IOS-XE-ntp-oper
1.3.6.1.2.1.197.1.2.2	/ntp-oper-data/ntp-status-info/stratum	Cisco-IOS-XE-ntp-oper
1.3.6.1.2.1.197.1.2.7	/ntp-oper-data/ntp-status-info/root-disp	Cisco-IOS-XE-ntp-oper
1.3.6.1.2.1.197.1.2.5	/ntp-oper-data/ntp-status-info/offset	Cisco-IOS-XE-ntp-oper
1.3.6.1.2.1.197.1.3.1.1.7	/ntp-oper-data/ntp-status-info/ntp-associations/peer-stratum	Cisco-IOS-XE-ntp-oper
1.3.6.1.2.1.197.1.3.1.1.9	/ntp-oper-data/ntp-status-info/ntp-associations/delay	Cisco-IOS-XE-ntp-oper
1.3.6.1.2.1.197.1.3.1.1.6	/ntp-oper-data/ntp-status-info/ntp-associations/offset	Cisco-IOS-XE-ntp-oper
1.3.6.1.2.1.197.1.3.1.1.8	/ntp-oper-data/ntp-status-info/ntp-associations/jitter	Cisco-IOS-XE-ntp-oper
1.3.6.1.2.1.197.1.3.1.1.5	/ntp-oper-data/ntp-status-info/ntp-associations/ntp-address/ip-addr	Cisco-IOS-XE-ntp-oper

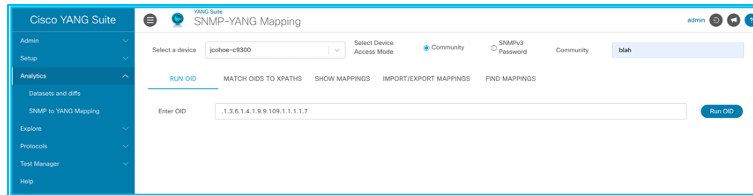
SNMP OID -> YANG Xpath Workflow



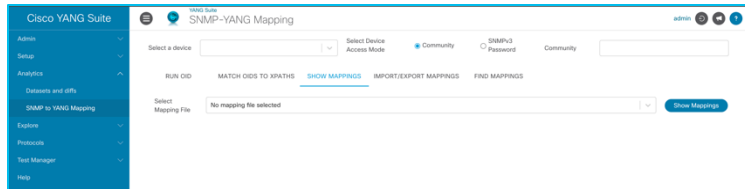
SNMP -> YANG Quick Guide

SNMP OID -> YANG Xpath Mappings are stored in CSV files

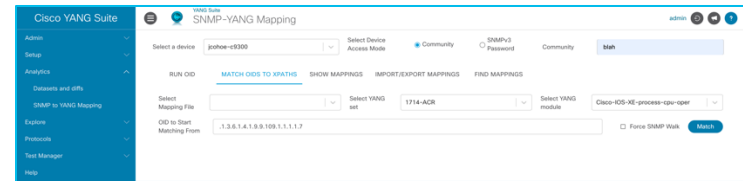
To check the output for an OID against a given device,
use RUN OID



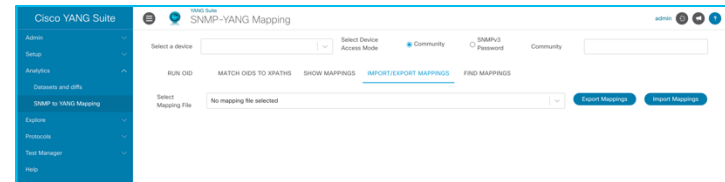
To show the mappings in a CSV file,
use SHOW MAPPINGS



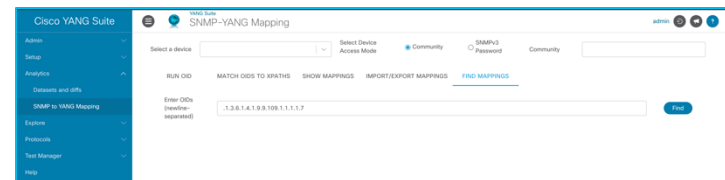
To compare the output for an OID and an Xpath,
use MATCH OIDS TO XPATHS



To import or export a CSV file,
use IMPORT/EXPORT MAPPINGS



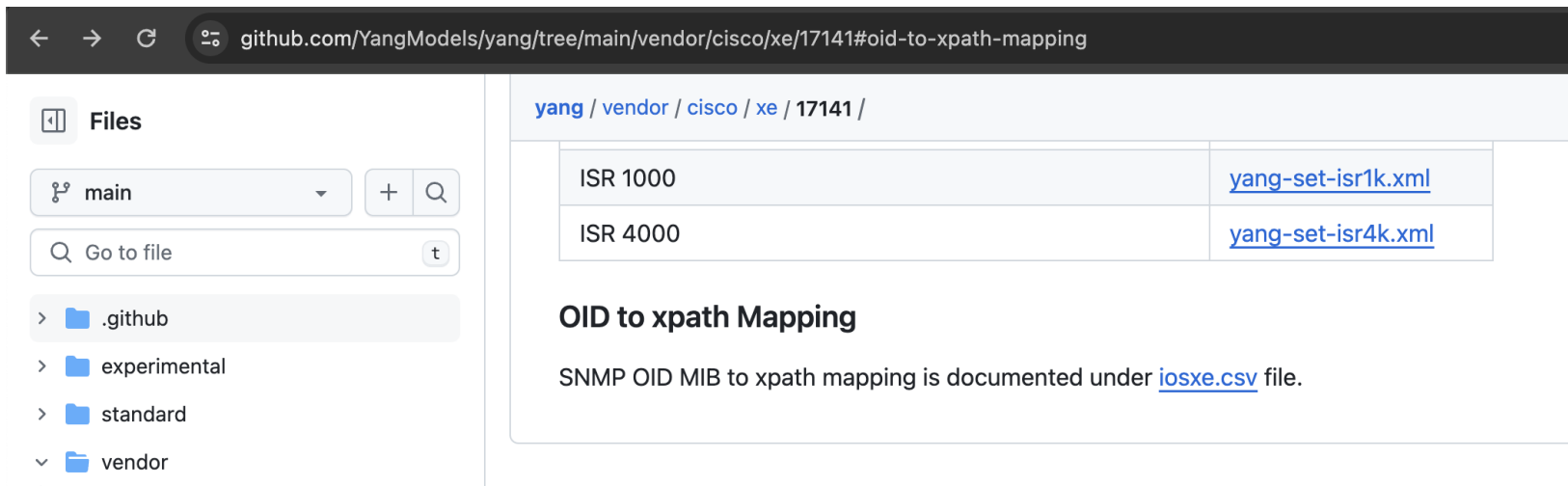
To find the mapping for one or more OIDs in all CSV files,
use FIND MAPPINGS



Mapping File available on GitHub

Starting in 17.14.1

Download the file from GitHub to load into YANG Suite



The screenshot shows a GitHub repository page for the file `yang/vendor/cisco/xe/17141#oid-to-xpath-mapping`. The left sidebar displays the file tree with folders `.github`, `experimental`, `standard`, and `vendor`. The main content area shows a table with two rows:

yang / vendor / cisco / xe / 17141 /	
ISR 1000	yang-set-isr1k.xml
ISR 4000	yang-set-isr4k.xml

Below the table, there is a section titled **OID to xpath Mapping** with the text: "SNMP OID MIB to xpath mapping is documented under [iosxe.csv](#) file."

<https://github.com/YangModels/yang/tree/main/vendor/cisco/xe/17141#oid-to-xpath-mapping>

Import the mapping file into YANG Suite

To find the updated mapping (CSV) file, navigate to <https://github.com/YangModels/yang/blob/main/ven.dor/cisco/xs/37151/iosxe-snmp-oid-xpath-mapping.csv>

1. Within the “IMPORT/EXPORT MAPPINGS” tab, complete the following:
2. Click “Import Mappings”
3. From your local machine, select the mapping file to upload and open that file
4. Notice that YANG Suite has successfully saved the mapping file
5. Now, you can explore the mappings that were previously created

The image is a composite of five screenshots from the Cisco YANG Suite interface, illustrating the process of importing a mapping file. The screenshots are annotated with green circles and numbers 1 through 5. Screenshot 1 shows the 'IMPORT/EXPORT MAPPINGS' tab with the 'Import Mappings' button highlighted. Screenshot 2 shows a file selection dialog with the 'Open' button highlighted. Screenshot 3 shows the 'Import Mappings' dialog with the 'Open' button highlighted. Screenshot 4 shows the 'Import Mappings' dialog with the 'Close' button highlighted. Screenshot 5 shows the 'Import Mappings' dialog with the 'iosxe.csv' file highlighted in the list.

The screenshot shows the Cisco YANG Suite interface with the 'IMPORT/EXPORT MAPPINGS' tab selected. The 'Import Mappings' button is highlighted with a green circle '5'.

Show mappings

Within YANG Suite, navigate to

1. Analytics > SNMP to YANG Mapping
2. Open the “SHOW MAPPINGS” tab
3. Select the newly uploaded mapping file & click “Show Mappings”
4. Once the mappings are loaded, click the black triangle to expand the OIDs

The screenshot displays the Cisco YANG Suite interface for 'SNMP-YANG Mapping'. The left sidebar (1) shows the navigation menu with 'Analytics > SNMP to YANG Mapping' selected. The main area (2) has the 'SHOW MAPPINGS' tab active. A mapping file 'iosxe-snmp-oid-xpath-mapping.csv' is selected in the 'Select Mapping File' dropdown (3), and the 'Show Mappings' button is visible. Below, the 'MIB Path' is set to '.iso' (4) and the 'YANG Path' is set to 'Select one YANG XPath'. At the bottom are 'Run', 'Save', and 'Delete' buttons.

Find Mapping of OID to Xpath

The screenshot shows the Cisco YANG Suite interface. The sidebar on the left is highlighted with a green circle and the number 1. The main panel has a tab labeled 'FIND MAPPINGS' highlighted with a green circle and the number 2. Below the tabs, there is a section for 'Enter OIDs (newline-separated)' with the text '1.3.6.1.2.1.197.1.2.5' and a 'Find' button highlighted with a green circle and the number 3. A pop-up window titled 'YANG Suite' is shown in the foreground, containing a table with the following data:

<input type="checkbox"/>	OID	Numeric OID	YANG Xpath
<input type="checkbox"/>	.iso.org.dod.internet.mgmt.mib-2.rtpSmpMIB.rtpSmpMIBObjects.rtpEntStatus.rtpEntStatusActiveOffset	1.3.6.1.2.1.197.1.2.5	/ntp-oper-data/ntp-status/info/offset
<input type="checkbox"/>	.iso.org.dod.internet.mgmt.mib-2.rtpSmpMIB.rtpSmpMIBObjects.rtpEntStatus.rtpEntStatusActiveOffset	1.3.6.1.2.1.197.1.2.5	/ntp-oper-data/ntp-status/info/offset

The table has a 'Close' button at the bottom right. A green circle with the number 4 points to the 'YANG Xpath' column in the table.

1. Analytics > SNMP to YANG Mapping
2. Open the “FIND MAPPINGS” tab
3. Provide the OIDs of interest & click “Find”
4. If any mappings for the OID exist in ANY mapping files, they will appear in the pop-up window! (In this example, there are two mappings because the mapping appears in more than one mapping file)
5. These mappings can be exported (see demo in next slide)



- Admin
- Manage users
- Manage plugins
- View logs
- Setup
- Analytics
- Explore
- Protocols
- Test Manager
- Help

Select a device

Select Device
Access Mode

Community

SNMPv3
Password

Community

- RUN OID
- MATCH OIDS TO XPATHS
- SHOW MAPPINGS
- IMPORT/EXPORT MAPPINGS
- FIND MAPPINGS

Enter OIDs
(newline-separated)

Find

Map SNMP OIDs to YANG Xpaths

1. Within the “MATCH OIDS TO XPATHS” tab, complete the following:
2. [OPTIONAL] provide a mapping file, if one exists
3. Select a YANG Set and a YANG Module in this case, “Cisco-IOS-XE-process-cpu-oper”
4. Provide an OID to map, in this case “1.3.6.1.4.1.9.9.109.1.1.1.1.7”
5. Click the blue “MATCH” button
6. Once the tree loads, try using different Xpaths to map to a sub point within the tree. Click “Run” to compare these two
7. Once you’re satisfied with a mapping, select “Save” (or “Delete”) to locally store the mapping you’ve just created!



YANG Suite Resources

YANG Suite Learning Lab

Detailed learning lab for all YANG Suite capabilities
YANG Suite container is part of the learning lab with



The screenshot displays the Cisco DevNet Learning Labs Center interface for the 'Introduction to YANG Suite' lab. The left sidebar contains a table of contents with 11 steps. The main content area is titled 'Introduction to YANG Suite' and includes instructions to launch the YANG Suite server and connect via a browser. A terminal window on the right shows the command `echo $DEVENV_APP_0880_URL` and its output: `https://yang-0880-20250816133740306c-devnet-rtg-us-west-1-2-devnet1.tandl.com/develop/yang-suite/`.

What's included:

- Access to YANG Suite
- Tutorials on getting started
 - Connect a device to YANG Suite
 - Download YANG models from device
- Explore YANG models
- NETCONF: GET/SET/SUBSCRIBE
 - Generate Ansible & Python
- RESTCONF: GET/SET/SUBSCRIBE
 - Generate Ansible & Python
- gNMI: GET/SET/Subscribe
 - Generate Ansible
- gRPC Telemetry receiver
- SNMP -> YANG mapping
- Datasets & Diffs

<https://developer.cisco.com/learning/labs/intro-yangsuite/>

YANG Suite Resources

Blogs

March 8, 2021

Developer
The Wait Is Over: Cisco YANG Suite Is Here!

Jeremy Cohee

This blog is authored by Jeremy Cohee and Kareem Iskander

As Cisco platforms (IOS XE, IOS XR, NX-OS) continue to support IETF, OpenConfig, and Cisco Native YANG models, we needed to develop a tool to help make the lives of network automation engineers easier!

<https://blogs.cisco.com/developer/363-yangsuite-01>

Developer
Free Tool Helps You Visualize and Understand YANG Models

Story DeViese

All YANG Suite, all the time

Are you interested in automating the process of viewing operational data or configuring network devices remotely? YANG models are the foundation to automation and programmability for Cisco IOS XE devices. Not sure where to start? Cisco YANG Suite is a free tool to help understand and visualize the Another Next Generation (YANG) models ranging from standards-based models such as OpenConfig and IETF to Cisco native models. Start using YANG Suite today to become a programmability pro!

<https://blogs.cisco.com/developer/yangalifetime01>

April 28, 2022

Developer
Snack Minute #58
YANG Suite updates and new features: A sneak peek

Matt DeNapoli

Have you tried out Cisco YANG Suite yet? YANG Suite provides network operators with a set of tools and plugins to interact with and test YANG programmable interfaces including NETCONF, RESTCONF, gNMI, and more.

In Episode 58 of DevNet Snack Minute, Jeremy Cohee (Technical Marketing Engineer with Cisco Systems) provides us with an overview of the new updates included in the latest version of Cisco YANG Suite.

<https://blogs.cisco.com/developer/2022yangsuiteupdatesfeatures01>

June 7, 2021

Developer
How Cisco Leverages YANG Suite for YANG Development

Jeremy Cohee

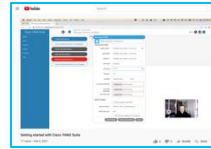
This blog is co-authored by Han Tieu, a Software Engineering Leader with Cisco

What is YANG Suite?

Cisco YANG Suite provides a set of tools and plugins to learn, test, and adopt YANG programmable interfaces. YANG Suite was released earlier this year, but it's no stranger to Cisco Engineering teams. It has already been leveraged within Cisco for many years for YANG model development.

<https://blogs.cisco.com/developer/averageyangsuite01?dtid=ossdc000283>

YouTube Videos



<https://youtu.be/smrhjLSAvz0>



<https://www.youtube.com/watch?v=dTun33611JA>



<https://www.youtube.com/watch?v=soyWPrOfIQs>

See the new updates and sneak peeks of new features with YANG Suite. Ep 58.

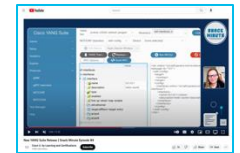


<https://www.youtube.com/watch?v=PkbAQz1vNk>



<https://www.youtube.com/watch?v=3zmNDfn8b38>

All YANG Suite, all the time, DevNet Snack Minute, Episode 9



https://www.youtube.com/watch?v=zVs0O9_6rAU

New YANG Suite Release | Snack Minute Episode 84

Additional Resources

<https://github.com/CiscoDevNet/yangsuite/>
<https://developer.cisco.com/yangsuite/>
<https://eurl.io/#MaW78CeIs> YANG Suite General (external)

Cisco YANG Suite – Resources Overview



- DevNet/YANGSuite: <https://developer.cisco.com/yangsuite>
- Documentation: <https://developer.cisco.com/docs/yangsuite>
- Gitub repository: <https://github.com/CiscoDevNet/yangsuite>
- Learning Lab: <https://developer.cisco.com/learning/lab/intro-yangsuite/step/1>
- PyPi: <https://pypi.org/project/yangsuite/>
- Brighttalk webinar: <https://www.brighttalk.com/webcast/17628/455586/the-wait-is-over-for-the-cisco-yang-suite-tooling>
- YANG Suite + NETCONF: <https://www.youtube.com/watch?v=dTun33611JA>

Questions? We are here to help.



YANG Suite Forum

Ask, collaborate, and share in the developer forum.



YANG Suite Support Mailer

Email the YANG Suite team with additional questions.

Quick start:

1. Clone: `git clone https://github.com/CiscoDevNet/yangsuite`
2. Build certificates: `cd yangsuite/docker/ ; ./gen_test_certs.sh`
3. Start container: `docker-compose up`
4. Connect and login: `http://localhost (admin/superuser)`

DevNet / YANG Suite

YANG Suite's home page is on DevNet – links and resources are here

DEVNET Discover Technologies Community Support Events New Announcement SIGN UP FREE LOG IN

Networking > YANG Suite Learn Docs Download

Cisco YANG Suite

Cisco YANG Suite provides a set of tools and plugins to learn, test, and adopt YANG programmable interfaces such as NETCONF, RESTCONF, gNMI and more.

YANG Suite will provides network operators with a common tool to interact with Cisco IOS XE, IOS XR, and the NX-OS Network Operating Systems as they look to modernize their network management and migrate from traditional network management tools.

[Read the docs](#)

Questions? We are here to help.



YANG Suite Forum

Ask, collaborate, and share in the developer forum.



YANG Suite Support Mailer

Email the YANG Suite team with additional questions.



Cisco Webex

Log in to get help right away in the Webex App.



Submit a Github Issue

Share and discuss enhancements in GitHub.

<https://developer.cisco.com/yangsuite>

Community Engagement



<https://community.cisco.com/>



yangsuite-support-ext@cisco.com

GitHub

<https://github.com/CiscoDevNet/yangsuite/issues>



<https://eurl.io/#MaW78CelS>

Documentation

<https://developer.cisco.com/docs/yangsuite>

The screenshot shows the Cisco DevNet website header with navigation links: Discover, Technologies, Community, Support, Events, and New Announcement. Below the header, the breadcrumb trail reads 'Documentation > YANG Suite'. The main content area is titled 'YANG Suite 1.0' and features a search bar. A left sidebar lists 'USER DOCUMENTATION' topics: Welcome to Cisco YANG Suite!, Managing YANG module files in YANG Suite, Device Profiles, Working with YANG Models, Using NETCONF with YANG Suite, YANG Suite Analytics, and Receiving gRPC telemetry with YANG Suite. The main content area is titled 'YANG Suite documentation' and 'User documentation', listing various topics such as 'Welcome to Cisco YANG Suite!', 'Device Profiles', 'Working with YANG Models', and 'Using NETCONF with YANG Suite'.

YANG Suite Documentation

- Search docs
- Welcome to YANG Suite!
 - Device Profiles
 - Managing device profiles
 - Setting Device to EnXR
 - Managing YANG module files in YANG Suite
 - Constructing and populating a YANG module repository
 - Uploading YANG files from the local filesystem to a YANG repository
 - Downloading YANG files via NETCONF from a device to a YANG repository
 - Copying YANG modules via SCP from a server to a YANG repository
 - Importing YANG modules from a Git repository to a YANG repository
 - Defining a YANG module set
 - Working with YANG Models
 - Exploring YANG Models
 - YANG Suite Analytics user documentation
 - Checking YANG model coverage
 - Datasets
 - Receiving gRPC telemetry with YANG Suite
 - Using NETCONF with YANG Suite
 - Using NETCONF RPCs
 - Working with NETCONF Notification Streams
 - Using Replays for repeated workflows
 - Locking and unlocking datastores
 - Setting Device to ENXR

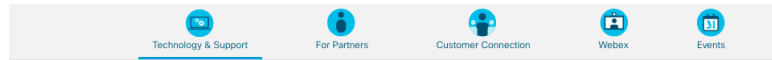
Built in the the GUI

<https://localhost/help>

A vertical navigation menu with a dark blue background and white text. The menu items are: Admin, Setup, Analytics, Explore, Protocols, and Help. Each item has a white downward-pointing chevron icon to its right.

Cisco Community

<https://community.cisco.com/t5/developer-networking/ct-p/j-developer-networking>



Cisco Community / Technology and Support / For Developers / Developer Networking

Developer Networking

NSO Developer Hub

LATEST POST - Re: inserting an entry to access-list via cli (cisco-ios-need)

Automation and Analytics

LATEST POST - Re: ZTP iDX and GuestShell on C9300

Cisco Digital Network Architecture (DNA)

LATEST POST - Re: DNA API Discovery

Embedded Networks

LATEST POST - networking

Physical and Virtual Network Elements

LATEST POST - RESTconf issues in IOS-XE

Other Networking Subjects

LATEST POST - Re: Philosophical Question on why are we turning to Programmability ?

Tools

LATEST POST - Re: Actools install help

YANG Tools

LATEST POST - YANG Suite label now available for developer discussions

Cisco Community / Technology and Support / For Developers / Developer Networking / YANG Tools

YANG Tools

The YANG Tools board hosts the discussions around YANG-related tools, such as the YANG Developer Kit (YDK), YANG Suite, and other tools for network automation.

Labels

YANG Development Kit (YDK) (571)

YANG Suite (1)



Navigate to the Developer Networking forum and select YANG Tools – There is a label for YANG Suite

Github.com/CiscoDevNet/yangsuite

Search or jump to... Pull requests Issues Marketplace Explore

CiscoDevNet / yangsuite Private

Watch 18 Star 1 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 2 branches 0 tags

Go to file Add file Code About

miott Merge pull request #7 from CiscoDevNet/develop 8cb57fe 4 days ago

docker	Added Cisco copyright to files.
docs	Added initial files including documentation
.gitignore	Added initial files including documentation
README.md	Added Cisco copyright to files.

Workflow to use

1. Git clone
2. Load certs
3. Docker build or pip install
4. Access through browser

Access docker with web browser

- Connect to <https://localhost> NOTE: You will get a complaint about the certs using "gen_test_certs.sh"
- Login using default user. You should change this after your first login.

Self-signed Certificate for testing

The SSL/TLS certificates installed with the nginx container should be obtained or generated using a trusted certificate authority. As a convenience you can use the yangsuite/docker/gen_test_certs.sh script to generate a self-signed certificate/key to get you going quickly.

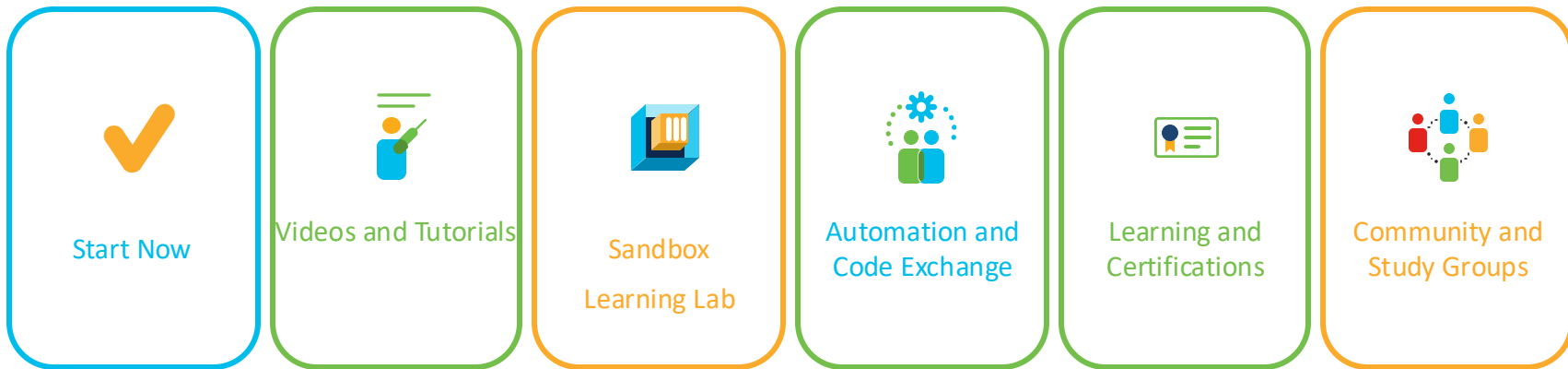
After obtaining or generating a certificate/key, run this command in the yangsuite/docker/ directory:

```
docker-compose up
```

The yangsuite/docker/yangsuite/production.py file contains Django settings.

Adding/Updating plugins can be done through the Admin-->Manage plugins page or by copying the plugin wheels into the yangsuite/docker/yangsuite/ directory, uncomment the appropriate lines in the yangsuite/docker/yangsuite/dockerfile, and run:

```
docker-compose up --build
```

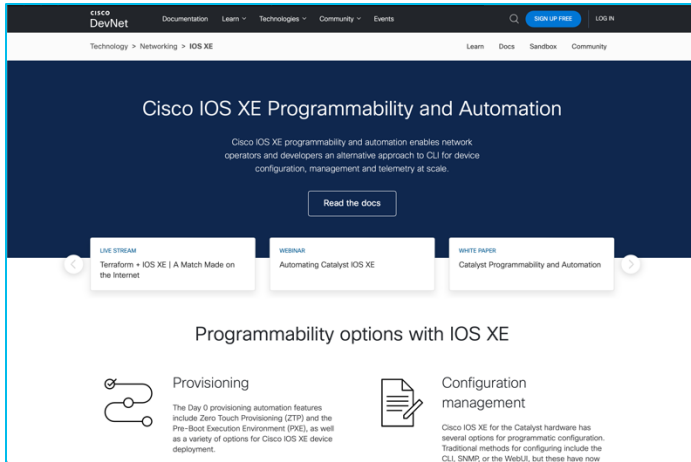


developer.cisco.com

IOS XE Developer Resources

Programmability Website

The one-stop-shop for Cisco IOS XE Programmability resources including videos, white papers, labs and more!



- Community Forum
- IOS XE FAQ
- White Papers
- Code Exchange
- IOS XE Docs & Guide
- Learning Tracks and Labs
- Sandboxes
- ... and more !



<https://developer.cisco.com/iosxe/>

API White Paper

Products & Services / Switches / Campus LAN Switches - Access / Cisco Catalyst 9300 Series Switches /

Catalyst Programmability and Automation

Programmability and auto... ^ 🔍

Table of Contents

Programmability and automatio... -

Day 0: Provisioning automation

Day 1: Model-driven programmability

Day 2: Model-driven telemetry

Day N: Device optimization

Cisco IOS XE operational consistency

Yet Another Next Generation (Y...

Day 1: Model-driven program...

Tooling: Cisco YANG Suite

Day 2: Model-driven telemetry

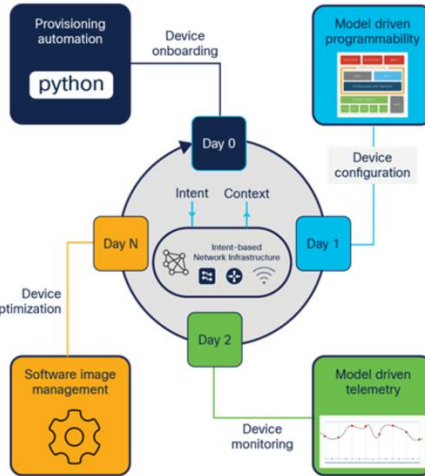
Day N: Device optimization

Conclusion

Additional resources

Blogs

Pre-boot Execution
Environment (iPXE)
Zero touch provisioning
VM automation



Network Configuration Protocol (NETCONF), RESTCONF, gNMI
YANG data models,
OpenConfig, and YANG
Suite tooling
Terraform, Ansible, pyATS

gNOI cert/os/reset proto
Guest shell + NETCONF
CentOS 8 Python 3
Application hosting
with Docker
CLI to YANG

TIG_MDT container + examples
YANG On-Change support
gRPC Dial-Out + DNS + TLS
gNMI/NETCONF Dial-In

<http://cs.co/apiw>



<http://cs.co/apiwpdf>

Website: <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/nb-06-catalyst-programmability-automation-wp.html>

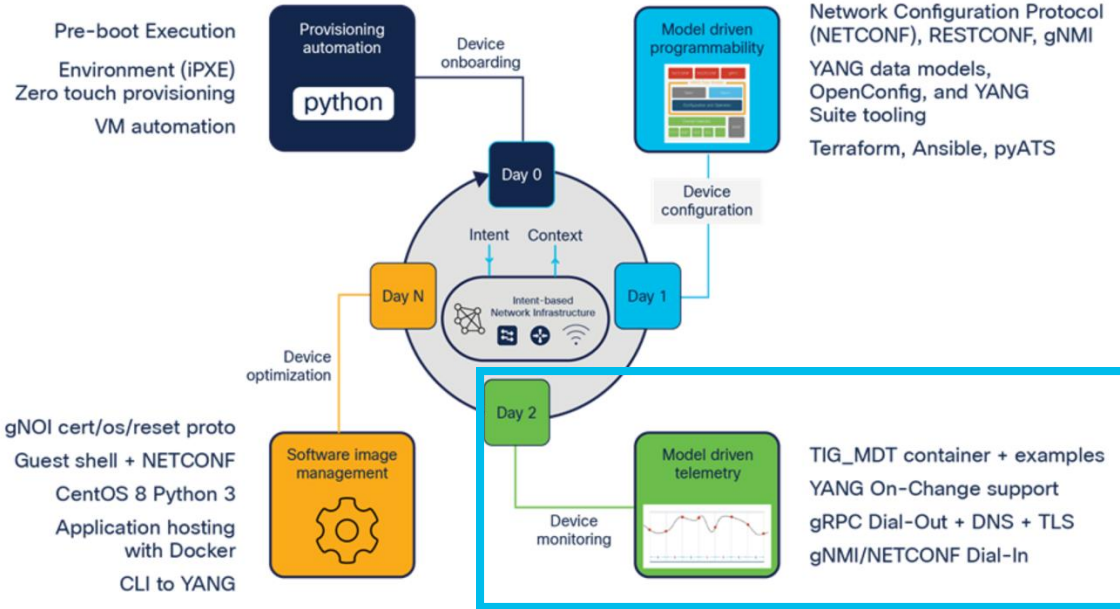
PDF: <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/nb-06-catalyst-programmability-automation-wp.pdf>

Webinar with live demos & examples: <https://www.youtube.com/watch?v=Ldck5PnPu2I>

MDT White Paper

The Model Driven Telemetry White Paper includes examples, use cases and tooling related to telemetry

- Introduction to Cisco IOS ...
- Table of Contents
 - Introduction to Cisco IOS XE
 - Introduction to telemetry
 - Benefits of model driven tele...
 - Network monitoring challeng...
 - Architecture and databases
 - Dial-in and dial-out MDT
 - Publication notification optio...
 - YANG data modeling language +
 - Benchmarking and comparis...
 - Cisco controller solutions
 - Cloud solutions
 - Tooling
 - Dashboarding and validation +
 - Configuration examples
 - Telemetry configuration man...
 - Troubleshooting and validati...
 - Best practices and lessons l...
 - Conclusion
 - Resources



Website: <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst9300-series-switches/model-driven-telemetry-wp.html>
PDF: <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/model-driven-telemetry-wp.pdf>



<http://cs.co/mdtwp>



<http://cs.co/mdtwppdf>



dCloud Programmability

<https://dcloud.cisco.com>

“Cisco Catalyst 9000 IOS XE Programmability & Automation Lab v1”

<https://dcloud2.cisco.com/demo/catalyst-9000-ios-xe-programmability-automation-lab-v1>

Use Cases:

EVPN:

Ansible with CLI deployment of EVPN solutions
 EVPN management over RESTCONF/YANG with Postman
 Declarative EVPN fabric management with Terraform

Model Driven Telemetry

Telemetry configuration with CLI and YANG Suite
 Collection with TIG_MDT container and tooling

YANG Programmability

YANG Suite tooling and integrations to YANG API's
 Ansible integrations

Tooling and Integrations

YANG Suite

- NETCONF/RESTCONF/gNMI API
 - Ansible integration
- NETCONF/gNMI Dial-In Telemetry
- gRPC Dial-Out Telemetry receiver

Telemetry

- TIG stack in Docker
- Grafana dashboard for device health

Postman / RESTCONF

- EVPN fabric API calls

Terraform/RESTCONF

- Declarative EVPN fabric management

Ansible

- EVPN solution enablement using CLI

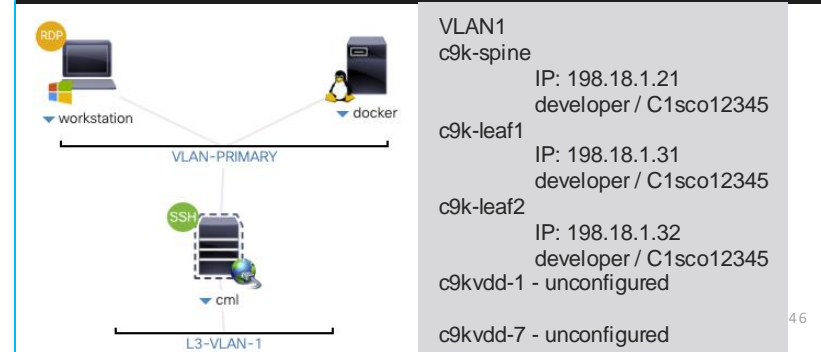
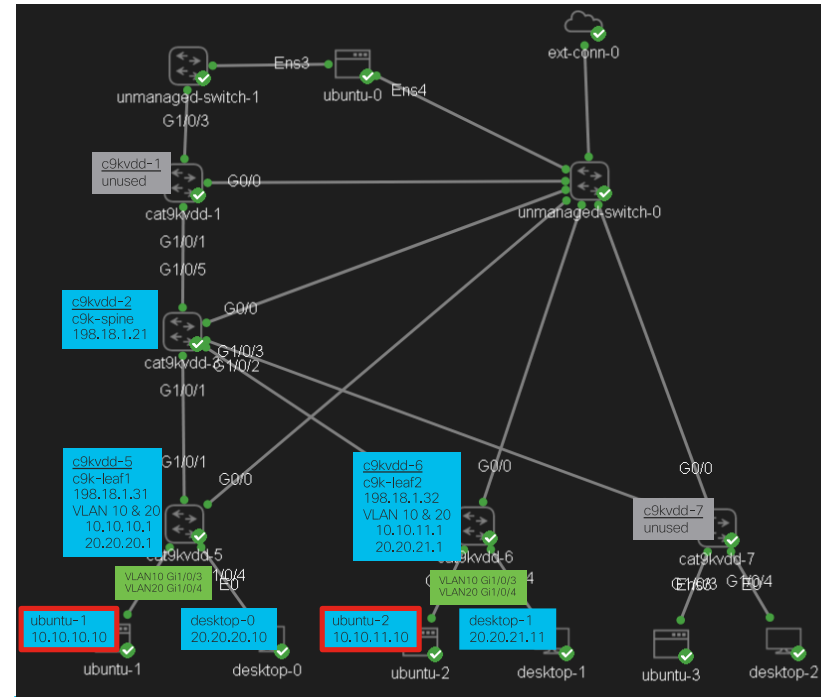
Ubuntu VM Details:

Syslog receiver from all switches
 TFTP config backup
 See slide

Windows VM Details

VS Code
 Terraform @ folder
 Ansible @ folder
 Chrome browser
 YANG Suite, Grafana
 Bash/PS/Cmd shells
 SSH into C9K or Ubuntu
 Postman
 Workspace for EVPN

C9K VM's



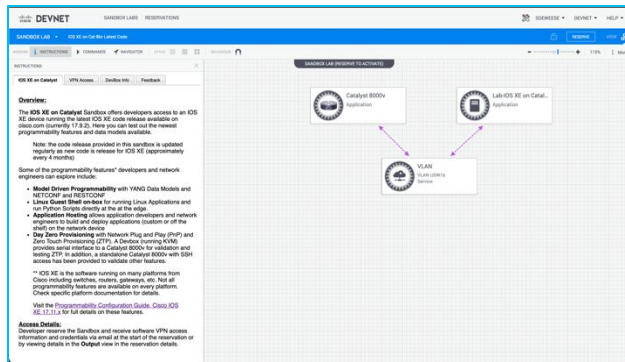
IOS XE Sandboxes

Easily access IOS XE as part of the DevNet Sandbox

This DevNet reservable sandbox has IOS XRv + N9Kv + IOS XE
The c8k within the DevBox is enabled for Day 0/ZTP usecases

<https://devnetsandbox.cisco.com>

“IOS XE on Cat8kv”



The “IOS XE on Cat8kv Always On” virtual sandbox requires no reservation

The c8kv can be accessed with SSH, NETCONF, and RESTCONF

Hostname: devnetsandboxiosxe.cisco.com

Username: admin Password: C1sco12345

SSH port 22, NETCONF port 830

devnetsandboxiosxe.cisco.com

“IOS XE on Cat8kv AlwaysOn”

Cisco IOS XE - Reservable Virtual Sandbox

This DevNet reservable sandbox has IOS XRv + N9Kv + IOS XE
The Catalyst 8000V within the DevBox is enabled for Day 0/ZTP usecases

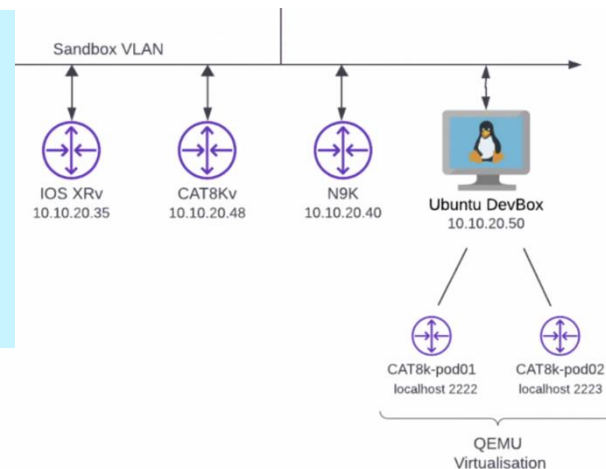
<https://devnetsandbox.cisco.com>

“IOS XE on Cat8kv”

<https://devnetsandbox.cisco.com/DevNet/catalog/IOS%20XE%20on%20Cat8kv>

The “Enterprise Networking” Learning Labs contains guides for the supported usecases

<https://developer.cisco.com/learning/>



Learning Track
Enterprise Networking

Explore YANG data models and NETCONF to query and configure IOS XE network devices. Then use Ansible to manage configurations on your IOS XE devices, covering the fundamentals of Ansible. Lastly, enable...

6 hrs - 4 modules - 12 labs

IOS XE on Cat8kv

The IOS XE on Catalyst 17.9.2 Sandbox offers developers access to an IOS XE device running the latest IOS XE code release available on cisco.com. Here you can test out the newest programmability...

Networking reservable



Sandbox Capabilities:

TIG_MDT Telemetry
YANG Suite API
Terraform + Ansible
ZTP & Guest Shell
... and more

Table of Contents

- Overview
- Network Topology
- Grafana Dashboard
- Access Details
- Additional Resources
- VPN Access

Catalyst 8000v Credentials

host	port	username	password
10.10.20.48	22	developer	C1sco12345
10.10.20.48	830	developer	C1sco12345
10.10.20.48	443	developer	C1sco12345

Developer Box Credentials

host	port	username	password
10.10.20.50	22	developer	C1sco12345
telnet localhost (CAT8k-1)	2222	developer	C1sco12345
telnet localhost (CAT8k-2)	2223	developer	C1sco12345

Cisco IOS XE - Always On Virtual Sandbox

The “[IOS XE on Cat8kv Always On](#)” virtual sandbox requires no reservation

The c8kv can be accessed with SSH, NETCONF, and RESTCONF

Hostname: sandbox-iosxe-latest-1.cisco.com

Username: admin Password: C1sco12345

SSH port 22, NETCONF port 830, RESTCONF HTTPS

<https://devnetsandbox.cisco.com>

devnetsandboxiosxe.cisco.com

<https://devnetsandbox.cisco.com/DevNet/catalog/ios-xe-always-on>

The “Enterprise Networking” Learning Labs contains guides for the supported usecases

<https://developer.cisco.com/learning/>

IOS XE on Cat8kv AlwaysOn



This AlwaysOn sandbox offers developers access to an IOS XE device running the latest IOS XE code release available on cisco.com (currently 17.11.x). Here you can test out the newest programmability feature...

Always-On



Launch

Access Details:

Developers and network engineers access the IOS XE on Catalyst 17.11.x sandbox directly using the following information:

- Cat8000v Host
 - Address: `sandbox-iosxe-latest-1.cisco.com`
 - Username: admin
 - Password: C1sco12345
 - NETCONF port: 830
 - gRPC telemetry port: 57500
 - ssh port: 22

Learning Track

Enterprise Networking

Explore YANG data models and NETCONF to query and configure IOS XE network devices. Then use Ansible to manage configurations on your IOS XE devices, covering the fundamentals of Ansible. Lastly, enable...

6 hrs - 4 modules - 12 labs

Programmability Configuration Guide

Preface

New and Changed Information

∨ Provisioning

Zero-Touch Provisioning

iPXE

∨ Shells and Scripting

Guest Shell

Python API

EEM Python Module

∨ Model-Driven Programmability

NETCONF Protocol

RESTCONF Protocol

NETCONF and RESTCONF Service-Level ACLs

gNMI Protocol

gRPC Network Operations Interface

gNMI Dial-Out Using the gRPC Tunnel Service

Model Based AAA

Model-Driven Telemetry

In-Service Model Update

∨ Application Hosting

Application Hosting

ThousandEyes Enterprise Agent



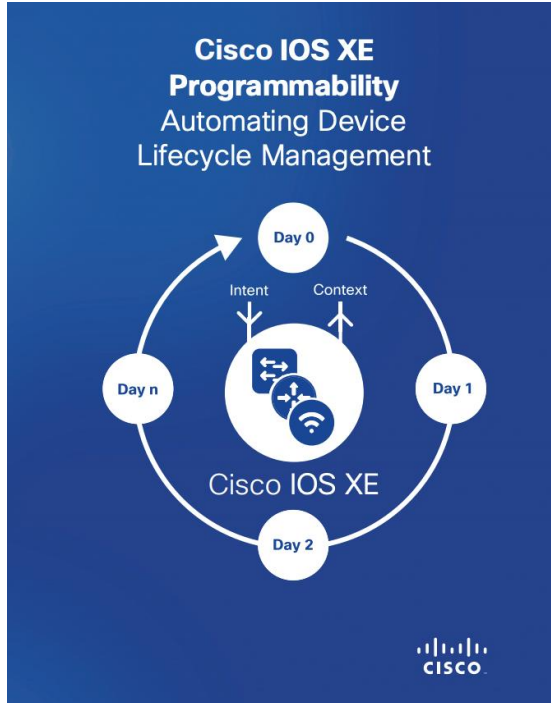
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1713/b_1713_programmability_cg.html

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1714/b_1714_programmability_cg.html

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1715/b_1715_programmability_cg.html



Cisco IOS XE Programmability – Resources



Programmability Configuration Guide, Cisco IOS XE Cupertino 17.9.x

First Published: 2022-08-01

- New and Changed Information
- ✓ Provisioning
 - Zero-Touch Provisioning
 - IPXE
- ✓ Shells and Scripting
 - Guest Shell
 - Python API
 - EEM Python Module
- ✓ Model-Driven Programmability
 - NETCONF Protocol
 - RESTCONF Protocol
 - NETCONF and RESTCONF Service-Level ACLs
 - gNMI Protocol
 - gRPC Network Operations Interface
 - Model Based AAA
 - Model-Driven Telemetry
 - In-Service Model Update
- ✓ Application Hosting
 - Application Hosting
- ✓ OpenFlow
 - OpenFlow
 - High Availability in OpenFlow Mode

<http://cs.co/programmabilitybook> OR <https://www.cisco.com/c/dam/en/us/products/collateral/enterprise-networks/nb-06-ios-xe-prog-ebook-cte-en.pdf>
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1715/b_1715_programmability_cg.html

Configure and Validate NETCONF Example Guide

This document describes how to configure and work with NETCONF and YANG on Cisco IOS XE using YANG Explorer

<https://www.cisco.com/c/en/us/support/docs/storage-networking/management/200933-YANG-NETCONF-Configuration-Validation.html>

1. Data Retrieval Example

Request a List of Interface Names from the Catalyst 3850

The desired operation can be selected from the left side Explorer section of the Yang Explorer application GUI. In this case, interface name data is to be retrieved from the Catalyst 3850 and so Oper (for operation) is selected followed by get-config under the interface name drop down. RPC is selected next in order to generate the YANG formatted (human readable) NETCONF RPC that is required to be sent to the Catalyst 3850 via NETCONF in order to retrieve this data from the Catalyst 3850.

The screenshot shows the Yang Explorer application interface. On the left, the 'Explorer' pane shows a tree view with 'leaf-interfaces' selected. Under 'leaf-interfaces', 'interface' is selected, and 'get-config' is chosen from the dropdown menu. The 'RPC' button is highlighted in red. The console area shows the following NETCONF message:

```
<?xml-stylesheet href="yang:ietf:params:xml:ns:netconf:base:1.0" type="text/css" />
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <name>
      <name>leaf-interfaces</name>
    </name>
  </get-config>
</rpc>
```

Status: Received HTTP Result for request type rpc

193

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Background Information

- Data Models - Programmatic and Standards Based Configuration and Monitoring
- Yet Another Next Generation (YANG) Data Modeling Language (RFC 6020)
- Network Configuration (NETCONF) Protocol (RFC 6241)

Configure

- Basic Configuration of a Catalyst 3850 Running IOS-XE 16.3.3 Software to Support NETCONF/YANG Data Modeling
- Additional (Optional) Configuration to Allow NETCONF/YANG Syslog and SNMP Event Monitoring
- Network Connectivity Configuration of the Catalyst 3850 Used in this Example

Verify NETCONF/YANG on the Catalyst 3850

Configure the Centralized Management Platform (Laptop)

- Install the Yang Explorer Application on a Laptop
- Use the Yang Explorer Application
- Subscribe to NETCONF Notifications (Optional)

Basic NETCONF/YANG Operational Examples

- Data Retrieval Example
- Request a List of Interface Names from the Catalyst 3850
- Configuration Example

Shut Down an Ethernet Interface on the Catalyst 3850

Catalyst 3850 CLI Display of the Interface Configuration both Before and After the Above NETCONF/YANG Configuration Change

Save the Configuration on a Catalyst 3850

Catalyst 3850 CLI Display of the Saved Startup Configuration After the Above NETCONF/YANG Configuration Save Operation

Configure the Catalyst 3850 from the CLI

3. Check What SNMP MIB Operational Data is Available via GET Request Operations

Load Additional YANG Data Models

- Load the Various YANG Data Model Files Individually
- Bulk Load of All the YANG Data Model Files at Once

Notable YANG Data Models

cisco-ia.yang Data Model

ned.yang Data Model

Python Scripting

Generate a Python Script from the Yang Explorer Application GUI

Run a Python Script from the Centralized Management Platform (Laptop)

Troubleshoot

NETCONF Error Messages

- RPC Error Example
- Other RPC Error Type Examples



The bridge to possible