



The bridge to possible

Cisco YANG Suite

YANG API Testing and Validation Tool

Technical Decision Maker

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

The top screenshot shows the 'Explore YANG Models' interface. It features a sidebar with navigation options: Admin, Setup, Explore, Protocols, and Help. The main area displays the 'Cisco-IOS-XE-interfaces-oper' module selected. A 'Node Properties' table is visible on the right:

Property	Value
Name	statistics
Node type	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

The bottom screenshot shows the 'NETCONF' interface. It includes a 'YANG Set' dropdown set to 'C9300' and a 'Module(s)' dropdown set to 'Cisco-IOS-XE-interfaces-oper'. Below these are buttons for 'Build RPC', 'Run RPC(s)', and 'Clear RPC(s)'. A 'Nodes' table is displayed:

Nodes	Value
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	

To the right of the nodes table is an XML RPC payload:

```
<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 IOS XE & YANG API Overview

2 YANG Suite Overview

3 Getting Started Workflow

4 Capabilities and Demos

5 Resources



Internal YANG Suite Resources

The focus of this presentation is for the public YANG Suite release

Internal Cisco Engineering users can refer to the resources below

Official Internal YANG Suite Wiki: <https://wiki.cisco.com/display/DDMICIA/Yang-Suite>

TL DR: Run the `setup_yangsuite.sh`

Internal Mailer: yang-suite-support@cisco.com

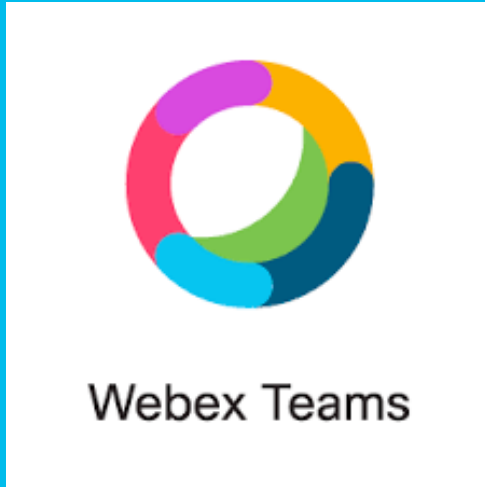
MPTE / YANG Suite internal support US - <https://eurl.io/#yaBfis2TI>

MPTE / YANG Suite internal support India - <https://eurl.io/#RqqTW49pG>

Mac OS X or Linux

Install Natively

Download install script [setup_yangsuite.sh](#)



Internal Programmability Teams Rooms

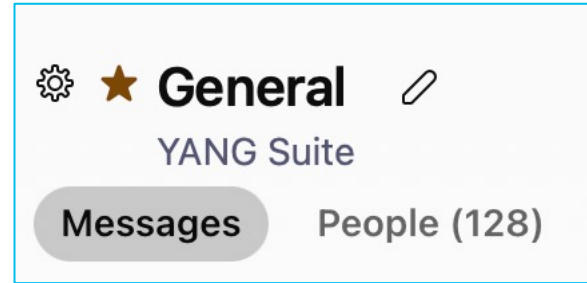
- Support for Sales and Engineering:
- NETCONF/YANG/GNMI + YangSuite + PyATS + XE/XR/NXOS - <https://eurl.io/#JY7rGIYda>
- IOX/Docker/App-Hosting/GuestShell/Python/ZTP on IOS XE - <https://eurl.io/#ryW56hQuS>
- Ask IBNG IOS XE Programmability - <https://eurl.io/#KJI4Yplbm>

- Support for Cisco internal developers:
- MPTE / YANG Suite internal support US - <https://eurl.io/#yaBfis2TI>
- MPTE / YANG Suite internal support India - <https://eurl.io/#RqgTW49pG>

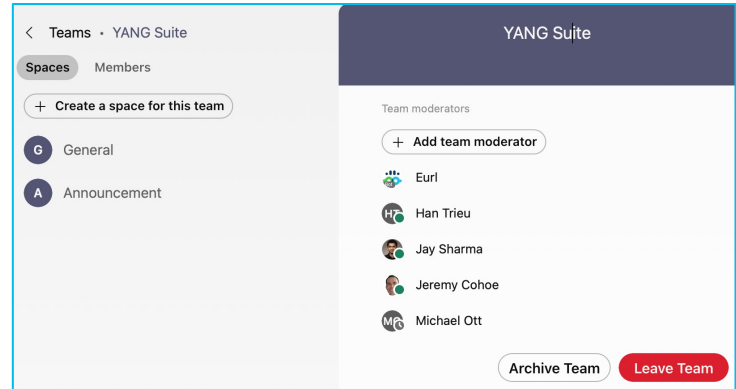


External Webex
Teams Rooms

Anyone can join !



<https://eurl.io/#MaW78CeIS>



Recent YANG Suite Innovations

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 [↗](#)

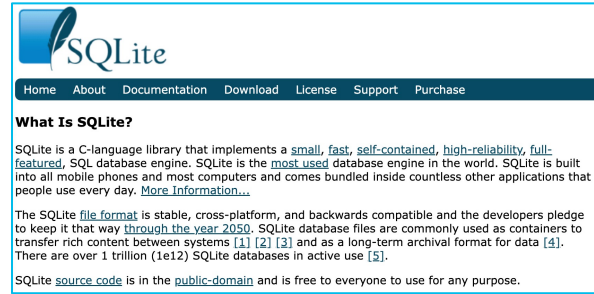
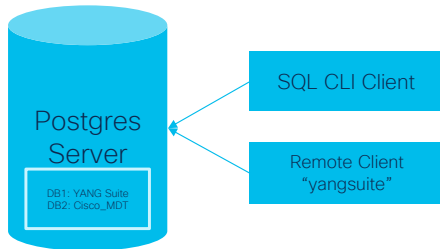
```
pip3 install --upgrade yangsuite yangsuite-devices yangsuite-filemanager yangsuite-yangtree yangsuite-coverage ya ↗
```

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

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#
```

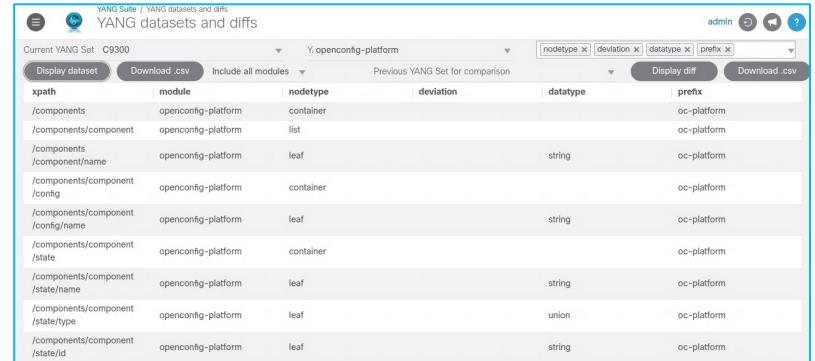
YANG datasets and diffs enhancements

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 'C9300' 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 comparison results with columns for 'xpath', 'module', 'nodetype', 'deviation', 'datatype', and 'prefix'.

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

Compare to previous:
17.9
17.9 Interfaces-Oper
17.9 Cisco-Native-*
Or OC, or IETF, etc...

YANG Suite +(mac) pip install + HTTPS support

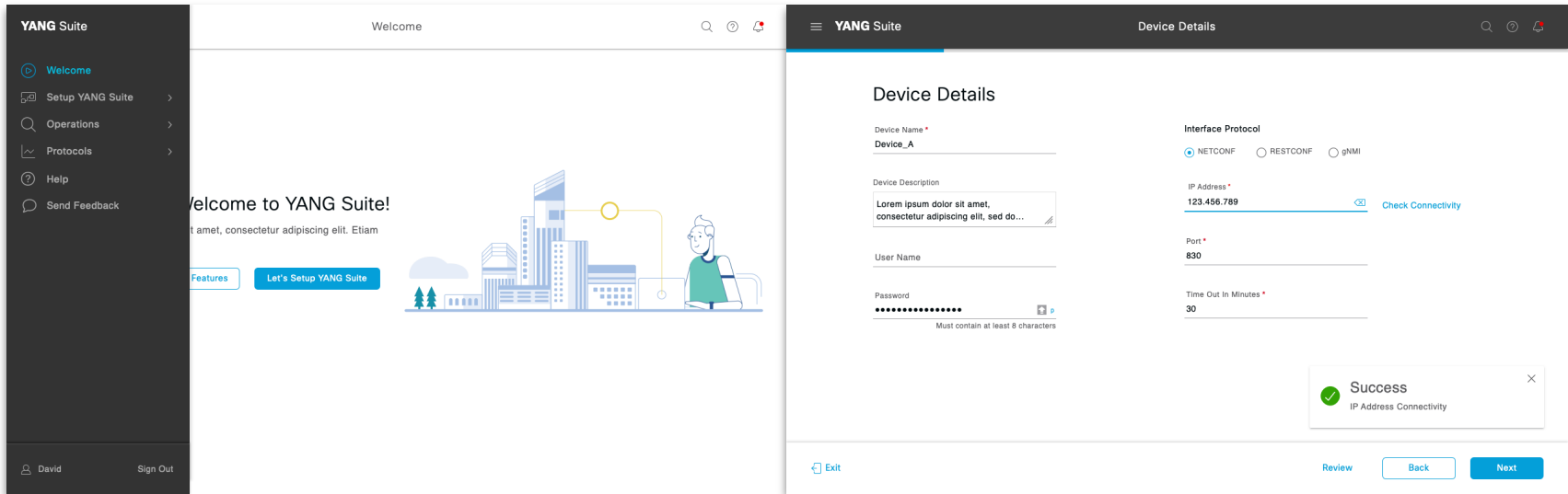
YANG Suite can now be accessed using HTTPS with TLS certificates when installed using the pip package manager mechanism.

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

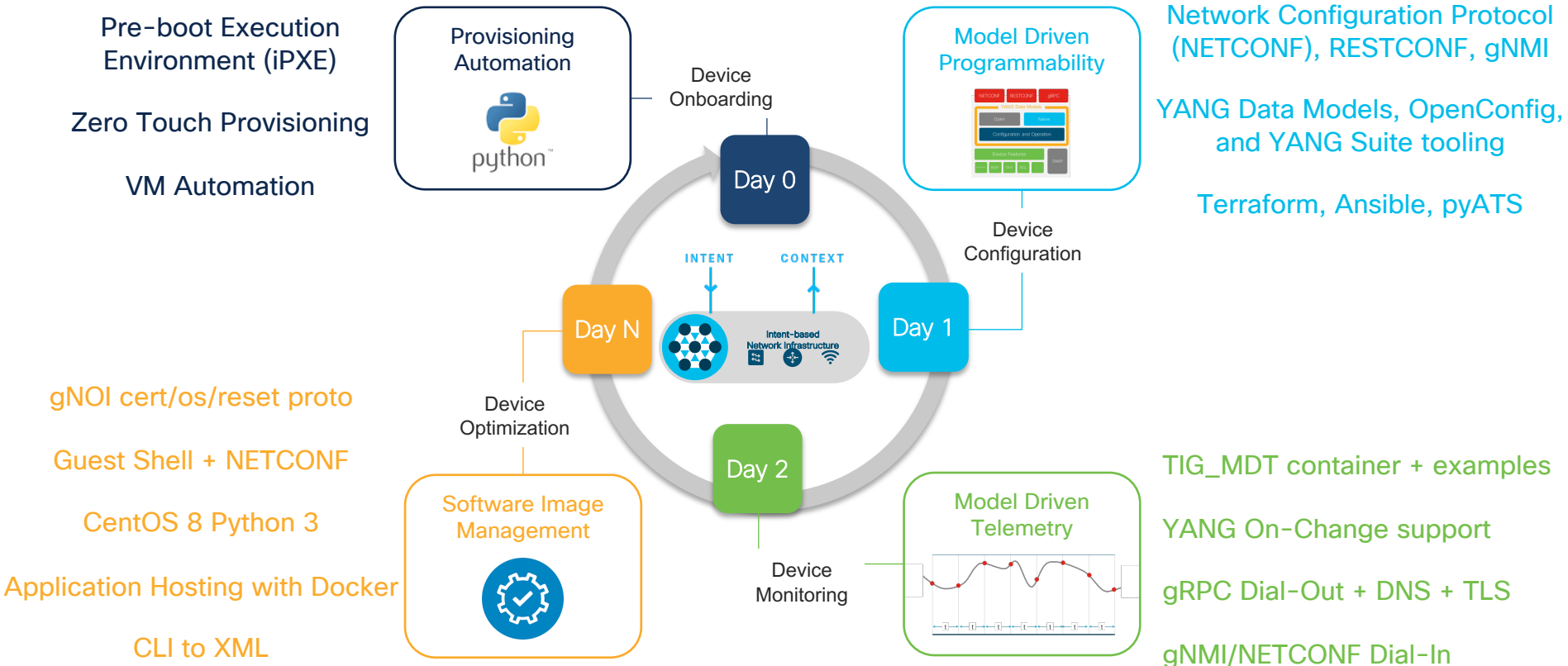
GUI UI/UX Redesign in progress

UI/UX Redesign underway – please provide any feedback via channels



IOS XE YANG API Overview

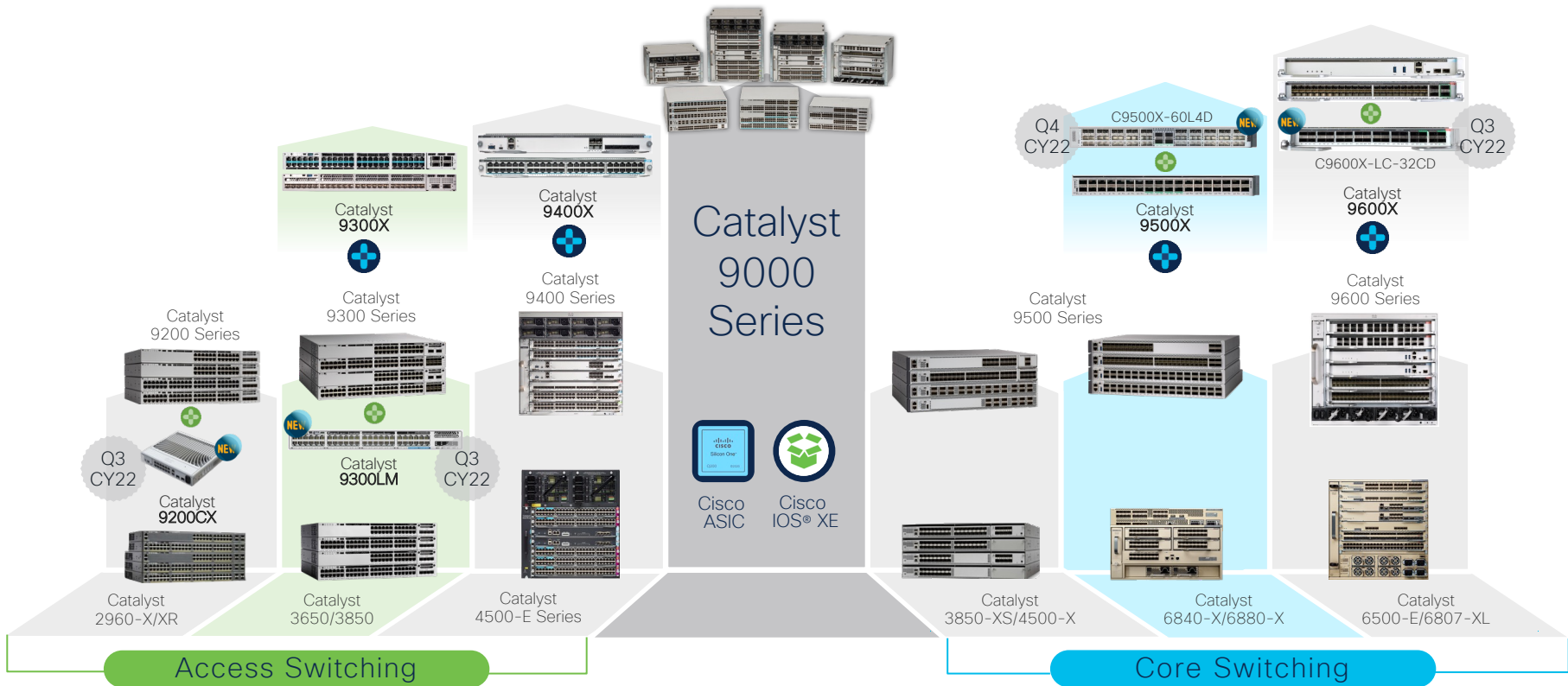
IOS XE Programmability & Automation Overview



Cisco Catalyst 9000 Switching Portfolio

One Family from Access to Core - Common Hardware & Software

2022 **NEW**



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_cg/configuring_yang_datamodel.html#id_84436

Demo Environment: IOS XE Always-On Sandbox

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

The screenshot shows the Cisco DevNet Sandbox Labs interface. At the top, the DevNet logo is on the left, and 'SANDBOX LABS' and 'RESERVATIONS' are in the center. A search bar on the left contains 'xe'. Below the search bar, a blue banner indicates 'LAB CATALOG (5)' with a 'Reset filters' link. The left sidebar has a 'Sandbox Labs' section with a gear icon, and a 'Reservations' section showing '0 New, 0 Total'. Below that, a 'FILTER BY:' section includes 'Sandbox Lab Status' with three options: 'Available', 'Unavailable', and 'View only', each with an unchecked checkbox. The main content area displays a list of lab cards. The first card is highlighted with a blue border and contains the following text:

Obtain the device connection parameters

Open a web browser tab to view the documentation for the [IOS XE on CSR Recommended Code Always-On Sandbox](#). Record the values for:

- The hostname (currently `sandbox-iosxe-latest-1.cisco.com`)
- The username (currently `developer`)
- The password (currently `Cisco12345`)
- The NETCONF port (currently `830`)

Below the card, three lab cards are visible:

- IOS XE on CSR Latest Code**
Explore the programmability capabilities of IOS XE on CSR1000v.
[RESERVE](#)
- IOS XE on CSR Latest Code ...**
Explore the programmability capabilities of IOS XE on CSR1000v.
[ALWAYS-ON](#)
- IOS XE on CSR Latest Code ...**
Zero Touch Provisioning (ZTP) and programmability on CSR1000v.
[RESERVE](#)

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

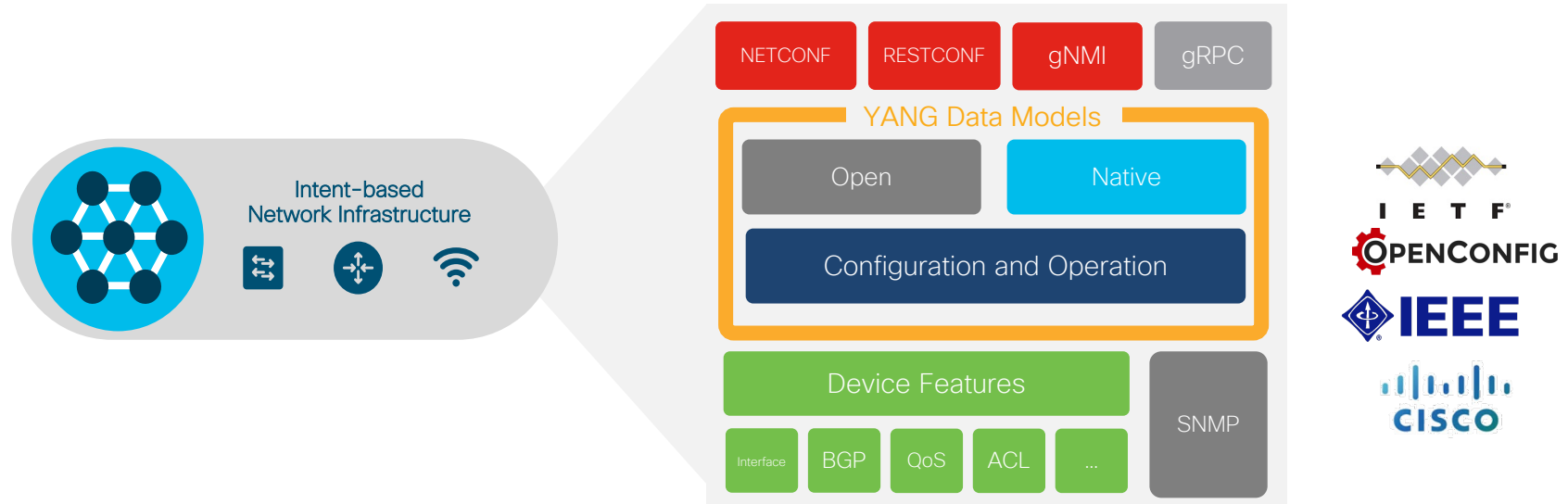
IOS XE YANG Model Driven Programmability

CLI

SNMP

WebUI

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

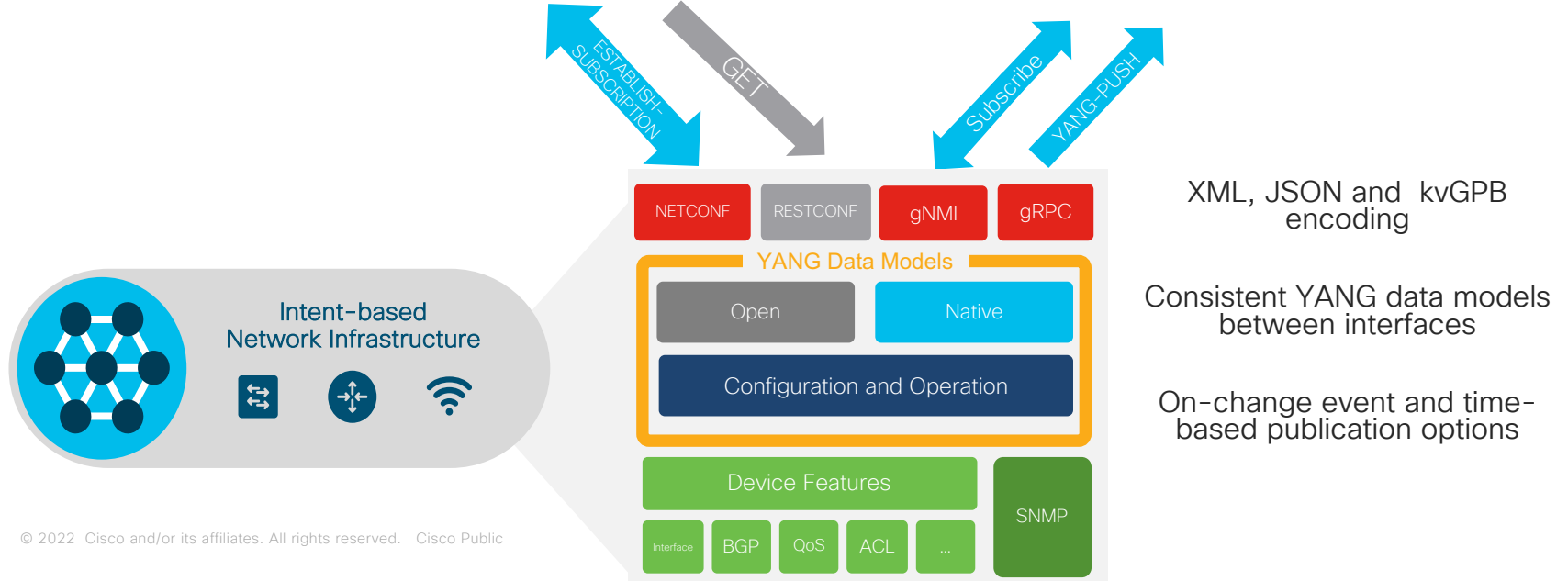


IOS XE Model Driven Telemetry Interfaces

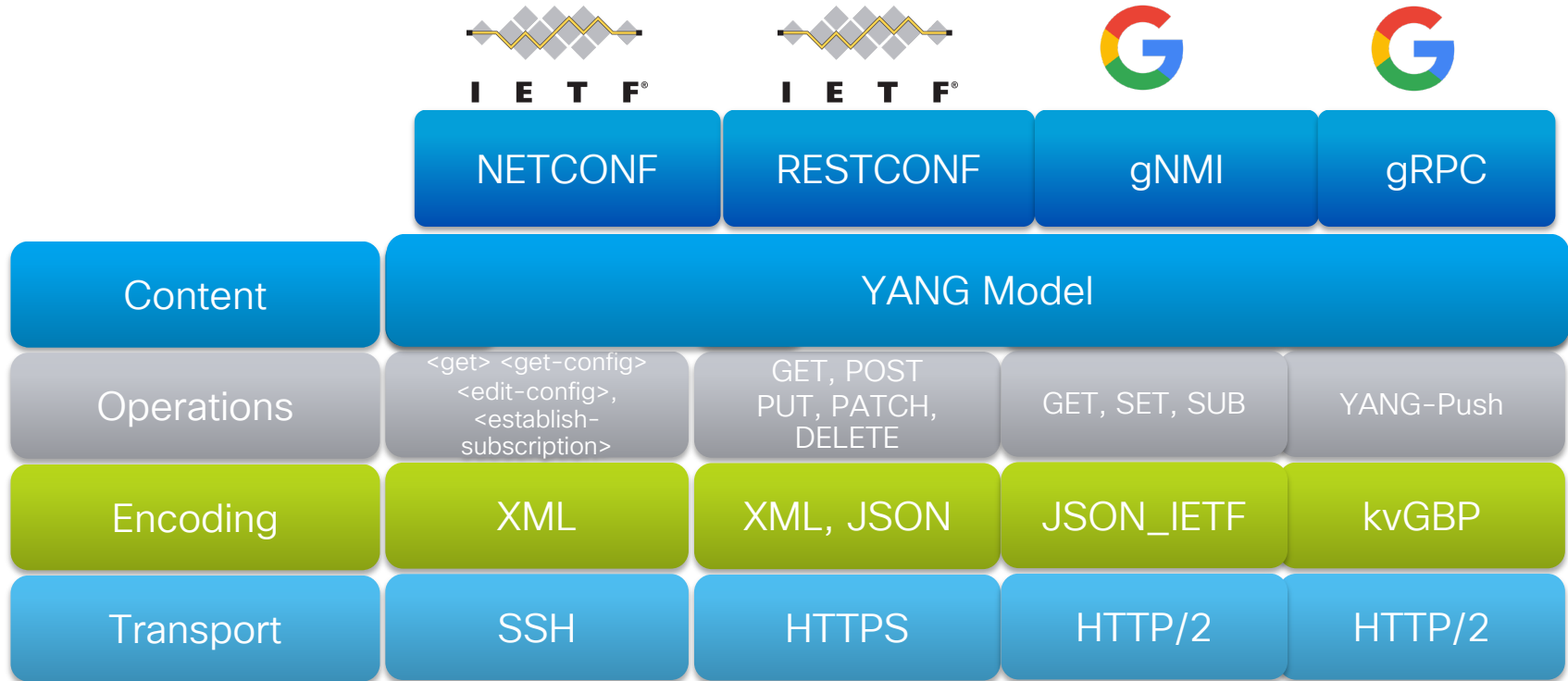
↔ Dial In: Session based - Collector establishes a connection to the device then subscribes to telemetry

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

Publication / Subscription

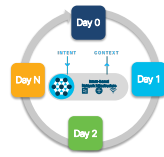


API Interfaces



API Operations

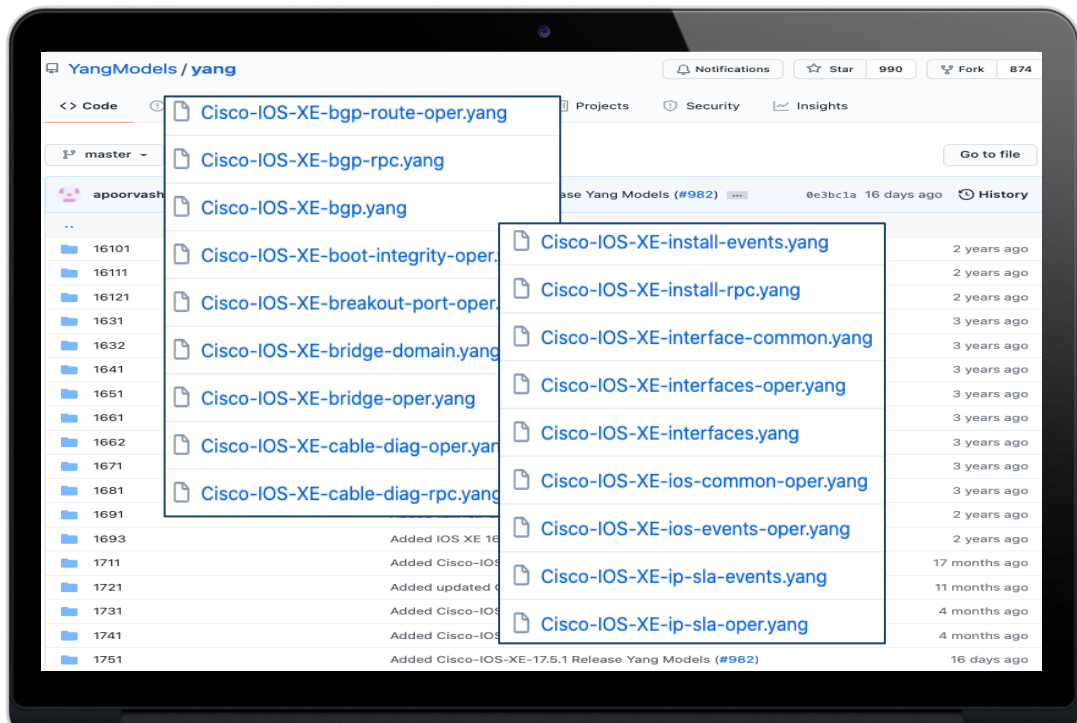
NETCONF	RESTCONF	gNMI	gRPC
<get-config>, <get>	GET	GET	
<edit-config> (operation="create")	POST	SET	
<edit-config> (operation="replace")	POST, PATCH	SET = update	
<edit-config> (operation="delete")	DELETE	SET = <null>	
<establish-subscription>		SUBSCRIBE	YANG push



IOS XE - YANG Model Coverage on GitHub

RFC7950 states that “YANG is a data modeling language used to model configuration data, state data, Remote Procedure Calls, and notifications for network management protocols”

YANG module name.yang	Description
Cisco-IOS-XE-native	running-config
Cisco-IOS-XE-{feature}-cfg	Feature configuration
Cisco-IOS-XE-{feature}-oper	Feature operational data
Cisco-IOS-XE-{feature}-rpc	Actions
Cisco-evpn-service	EVPN service abstraction
OpenConfig-{feature}	abstraction for config & oper

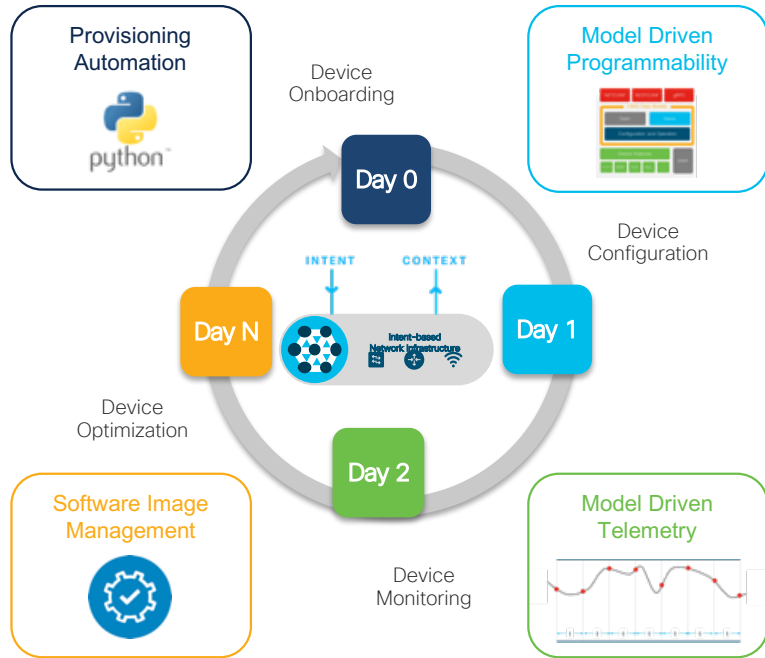


<https://github.com/YangModels/yang/tree/master/vendor/cisco/>

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

Cisco YANG Suite

Admin

- Manage users
- Manage plugins
- View logs

Setup

- YANG files and repositories
- YANG module sets
- Device profiles

Analytics

- Datasets and diffs
- SNMP to YANG Mapping

Protocols

- gNMI
- gRPC telemetry
- NETCONF
- RESTCONF

Explore

- YANG

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":

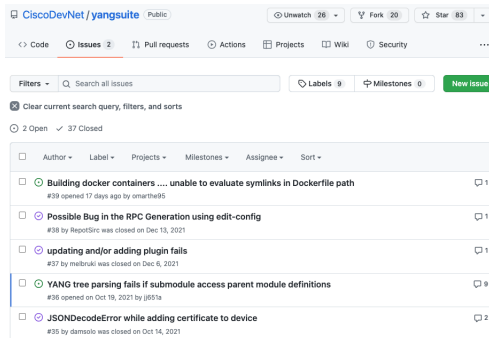
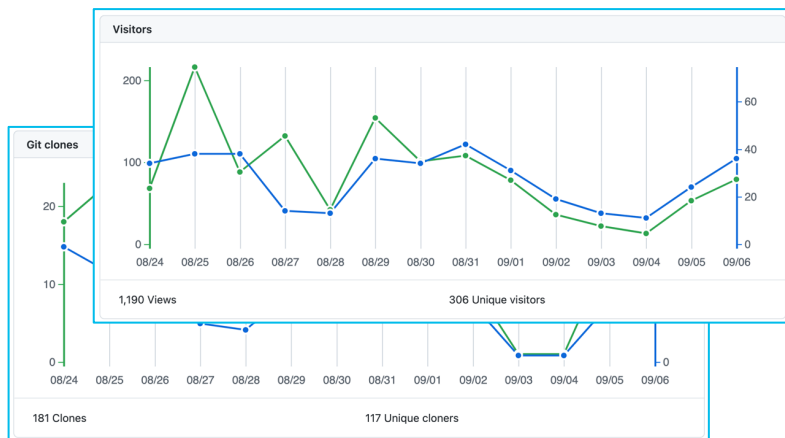
Decline Accept

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

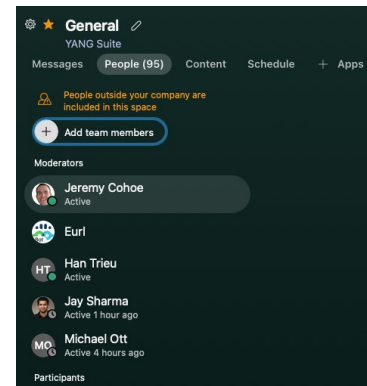
Decline Accept

YANG Suite Adoption

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/#MaW78CelS>



~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

The screenshot shows the 'YANG Suite / Plugins' interface. At the top, there's a navigation bar with a hamburger menu, a user icon, and the text 'YANG Suite / Plugins' and 'YANG Suite plugins'. On the right, there are icons for 'admin', a refresh button, a search icon, and a help icon.

The main content area is divided into two sections:

- Core YANG Suite plugins:** A table listing core plugins with columns for Package name, Description, Installed version, and Latest version.
- Installed optional plugins:** A table listing optional plugins with columns for Package name, Description, Installed version, and Latest version.

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

The screenshot shows the 'Cisco YANG Suite' interface. On the left, there's a sidebar with navigation options: Admin, Manage users, Manage plugins, View logs, Setup, YANG files and repositories, YANG module sets, Device profiles, Analytics, Explore, Protocols, and Help.

The main content area is titled 'YANG Suite / Device profiles' and 'Manage device profiles'. It has a 'Select a device profile' section with a radio button for 'sandbox-iox2e-latest-1.cisco.com'. Below that, there's a 'New Device Profile' form with a dropdown menu and a list of fields: Profile Name, Description, Address, Username, Password, and Timeout. A note indicates that fields marked with an asterisk are required.

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 via NETCONF. 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 'Cisco-IOS-XE-interfaces-oper' model with a tree structure including 'interfaces', 'interface', and 'interface-type'.
- gRPC Telemetry:** A section for listening for Model Driven Telemetry. It shows the current IP address as '127.0.0.1' and the port as '57344'. It includes a 'Stop telemetry receiver' button and a 'Clear output' button. The output shows the received telemetry data for the path '/five-seconds'.

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, origin, 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](#) [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...](#) [Show Legend](#) [Build JSON](#) [Clear Values](#) [Run RPC\(s\)](#) [Clear RPC\(s\)](#)

Nodes	Value	Operation
Cisco-IOS-XE-interfaces-oper		
interfaces	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
interface	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
name	<input type="text" value="*"/>	
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 container supporting HTTPS

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



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.

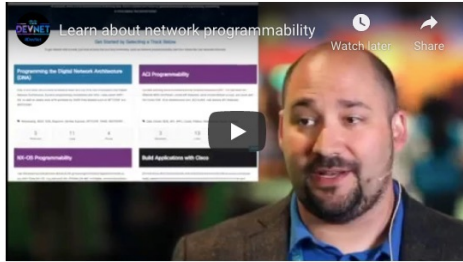
Doing the "git clone" has an issue with CRLF and you can workaround it with the following command in the gitshell:

```
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":

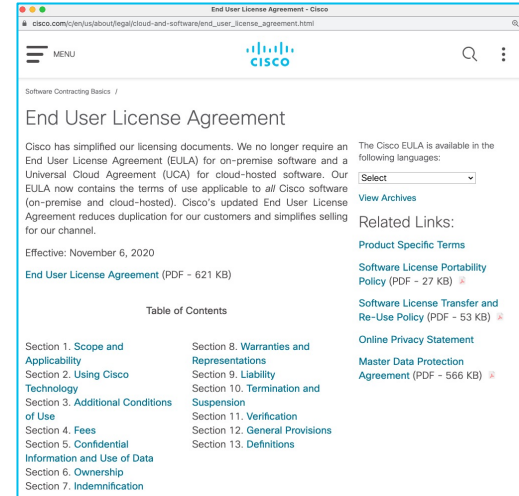
Decline Accept

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

Decline Accept

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

Log in to YANG Suite

Please login to access this page.

Username:

Password:

[Lost your password?](#)

YANG Suite v3 and Docker

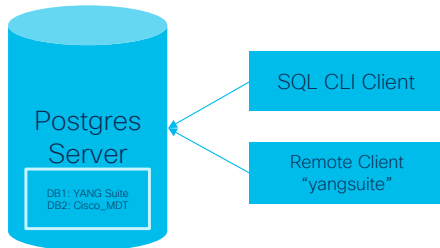
Release 3 brings enhancements to the Docker environment including:

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



SQLite

Home About Documentation Download License Support Purchase

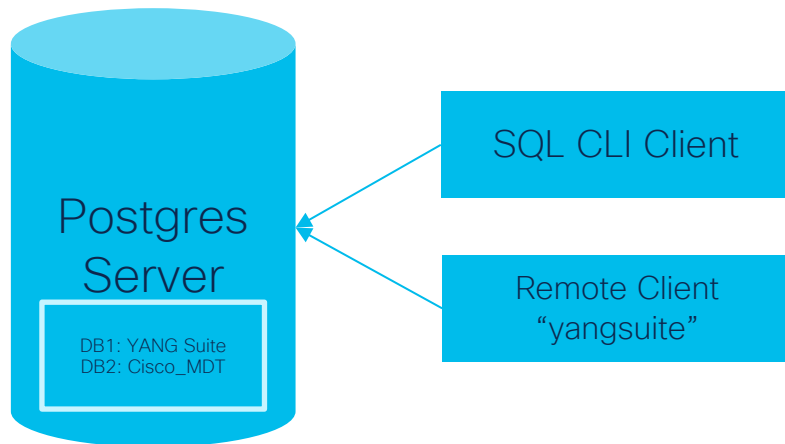
What Is SQLite?

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day. [More Information...](#)

The SQLite [file format](#) is stable, cross-platform, and backwards compatible and the developers pledge to keep it that way [through the year 2050](#). SQLite database files are commonly used as containers to transfer rich content between systems [\[1\]](#) [\[2\]](#) [\[3\]](#) and as a long-term archival format for data [\[4\]](#). There are over 1 trillion (1e12) SQLite databases in active use [\[5\]](#).

SQLite [source code](#) is in the [public-domain](#) and is free to everyone to use for any purpose.

```
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.

YS on Mac with Pip

Install Homebrew

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

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:

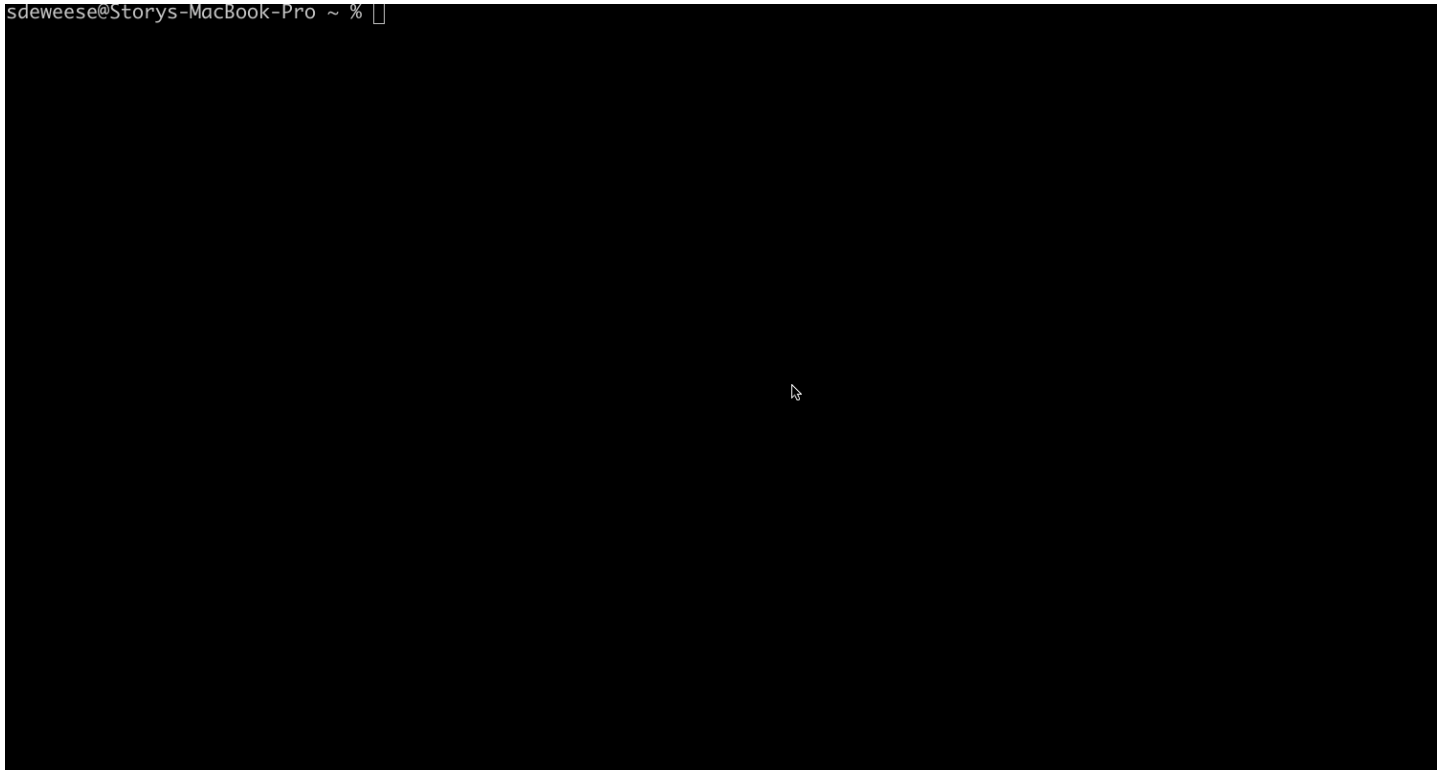
The path to the data is absolute so /Users/username_here/yangsuiteenv/ is recommended

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 ~ %
```



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 Cisco YANG Suite web interface. A red banner at the top contains the URL <https://localhost:8443>. The page title is "Welcome to Cisco YANG Suite!". A left-hand navigation menu is visible with items: Setup, Explore, Protocols, and Help. The main content area includes a welcome message, a brief description of the suite, and a section titled "Docker-based Installation" which lists the steps performed by the `yangsuite/docker/start_yang_suite.sh` script.

The screenshot shows the Cisco YANG Suite web interface. A red banner at the top contains the URL <http://localhost:8480>. The page title is "Welcome to Cisco YANG Suite!". A left-hand navigation menu is visible with items: Setup, Explore, Protocols, and Help. The main content area includes a welcome message, a brief description of the suite, and a section titled "Docker-based Installation" which lists the steps performed by the `yangsuite/docker/start_yang_suite.sh` script.

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 & Demos

- 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

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

Create new device
Check selected device
Clone selected device
Edit selected device
Delete selected device
Create default Repository

New Device Profile

Fields marked with * are required.

General Info

Profile Name * c9300
Description
Address * 10.1.1.5
Username admin
Password *****
Timeout * 30

Variables

Variable Name	Value
---------------	-------

TLS Authority Certificate: Choose File No file chosen
TLS Client Certificate: Choose File No file chosen
TLS Client Key: Choose File No file chosen
TLS Server Certificate: Choose File No file chosen
TLS Server Key: Choose File No file chosen

Create Profile Check Connectivity Cancel

New Device Profile

gNMI

Device supports gNMI

Platform * IOS XE
gNMI insecure port * 50052
gNMI secure port * 9339
 Use TLS Certificate
TLS host override

NETCONF

Device supports NETCONF

Device Variant * (Default - RFC-compliant device)
NETCONF port * 830
 Skip SSH key validation for this device
Address 10.1.1.5
Username admin
Password *****
Timeout 30
NETCONF port * 830
 Skip SSH key validation for this device
Address 10.1.1.5
Username admin
Password *****
Timeout 30

RESTCONF

Device supports RESTCONF

HTTP or HTTP(secure) encoding https
RESTCONF base URL /restconf
RESTCONF port * 443
Address 10.1.1.5
Username admin
Password *****

SSH

Device allows SSH login

Device variant * generic_termserver
Address 10.1.1.5
SSH Port * 22
Delay Factor 1.0
Username admin
Password *****
Timeout 30
 Use SSL Certificate

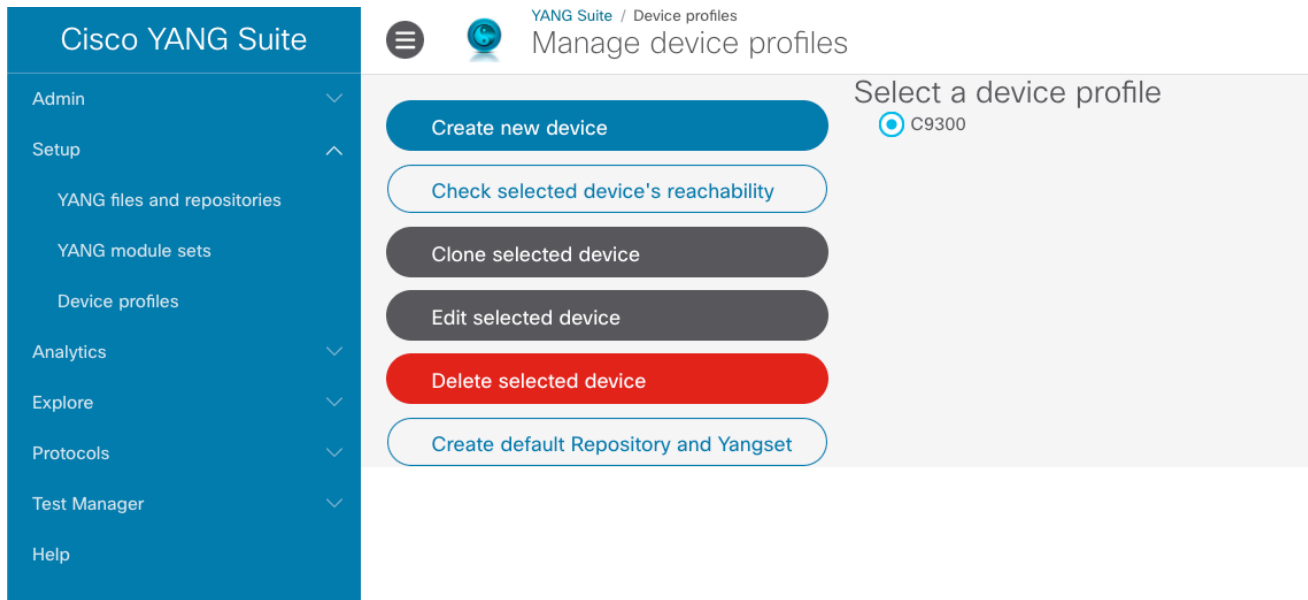
Create Profile Check Connectivity Cancel

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 selected for "C9300". Below this are several action buttons: "Create new device" (blue), "Check selected device's reachability" (light blue), "Clone selected device" (dark grey), "Edit selected device" (dark grey), "Delete selected device" (red), and "Create default Repository and Yangset" (light blue). A popup window titled "Connectivity check results for 'c930...'" is overlaid on the right, showing a list of checked items: ping, gNMI, NETCONF, and RESTCONF, each preceded by a green checkmark.

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

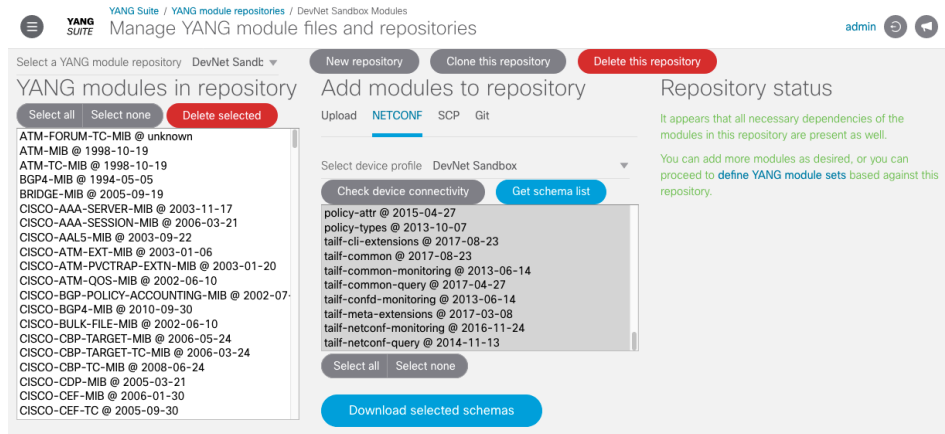
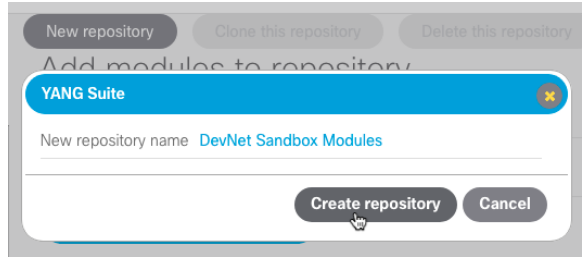
Create new device
Check selected device's reachability
Clone selected device
Edit selected device
Delete selected device
Create default Repository and Yangset

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

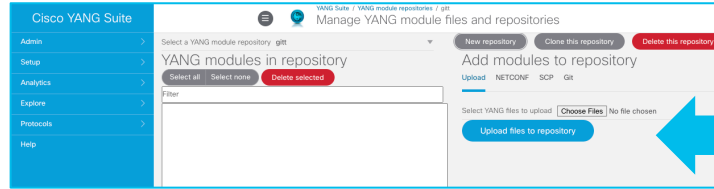
Select Setup > YANG Files and Repository

Create/Select YANG Repository

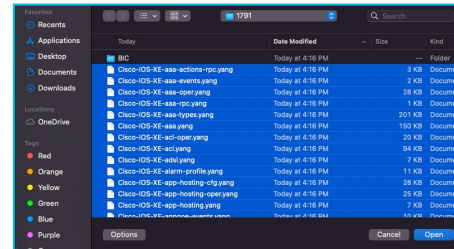
Navigate to “Upload” tab

Select the necessary YANG files.

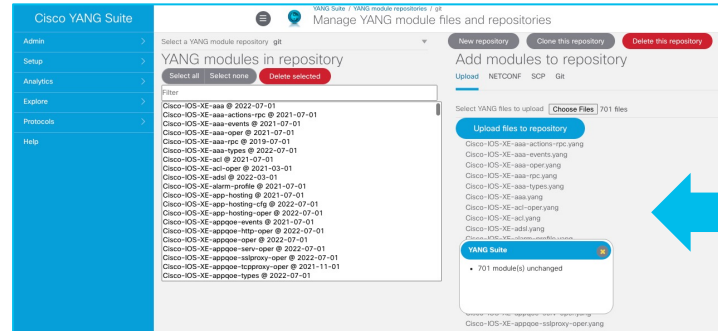
Select “Upload files to repository”.



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

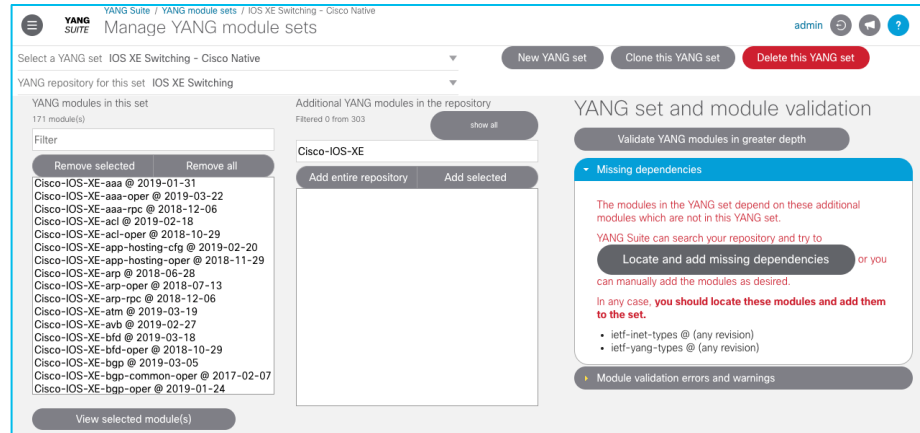
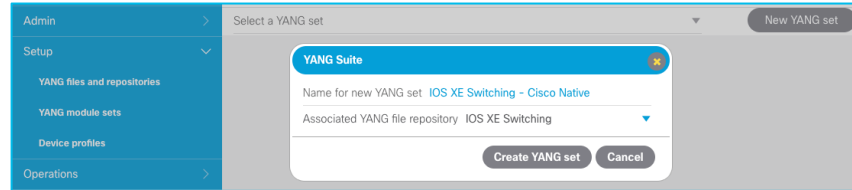
Select all > “Add selected”

Select “Locate and add missing dependencies”

Validation complete

This YANG set is currently empty.

Please add one or more YANG modules to this set.



- Explore

Explore YANG

Operations > Explore YANG
Select YANG Set and module then Load module
Details about the data are displayed

The screenshot shows the 'YANG Suite' interface. The top navigation bar includes 'YANG Suite', 'Exploring YANG', and 'YANG set "C9300-17.1" / Modules'. The main content area is titled 'Explore YANG Models' and features a search bar and a 'Load module(s)' button. Below this, there are search filters and a tree view of YANG models. The tree view shows a hierarchy starting with 'interfaces', followed by 'interface', and then various sub-nodes like 'name', 'interface-type', 'admin-status', etc. The 'statistics' node is highlighted. To the right of the tree view, the 'Node Properties' pane is open, displaying details for the 'statistics' node, including its name, node type, description, module, revision, XPath, prefix, namespace, access, operations, and a reference URL.

The 'Tree Icon Legend' section contains the following items:

- notification**: icon of a gift box
- output**: icon of a blue arrow pointing right
- rpc**: icon of an envelope
- submodule**: icon of a gear
- action**: icon of a target symbol
- anydata**: icon of a yellow asterisk
- anyxml**: icon of a green code symbol
- case**: icon of a blue arrow pointing right
- choice**: icon of a blue arrow pointing left
- container**: icon of a blue folder
- input**: icon of a blue arrow pointing right
- leaf**: icon of a green leaf
- leaf-list**: icon of a green leaf with a vertical line
- list**: icon of a blue list icon
- module**: icon of a gear with a plus sign

The 'Node Support' section contains the following items:

- status: deprecated**: icon of a yellow leaf
- status: obsolete**: icon of a red leaf
- deviation: not supported**: icon of a grey leaf with a red slash

The 'Node Badges' section contains the following item:

- list key**: icon of a red 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 'Cisco-IOS-XE-process-cpu-oper'. The left sidebar shows a tree view of the model, with 'five-seconds' selected under 'cpu-usage/cpu-utilization'. The right pane displays the 'Node Properties' 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

Admin Setup Analytics Explore YANG Protocols Test Manager Help

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 XPaths 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
 - lag-aggregate-state
 - ipv4-tcp-adjust-mss
 - ipv6-tcp-adjust-mss
 - intf-ext-state-support
 - intf-ext-state

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)

Demo: Datasets and Diffs

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

The screenshot shows the 'YANG datasets and diffs' interface. At the top, it indicates the current YANG Set is 'C9300' and the selected module is 'Y. openconfig-platform'. There are buttons for 'Display dataset', 'Download .csv', and 'Include all modules'. Below the table, there are buttons for 'Display diff' and 'Download .csv'. The table lists various YANG nodes with their paths, modules, node types, deviations, datatypes, and prefixes.

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 shows 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 displays the 'YANG datasets and diffs' tool. It shows the current YANG Set as '1711-c9300' and the YANG module as 'openconfig-lacp'. There are buttons for 'Display dataset', 'Download .csv', and 'Include all modules'. Below, it shows the previous YANG Set for comparison as 'jcohoe-c9300-2-default-yangset' with buttons for 'Display diff' and 'Download .csv'. A table displays the comparison results:

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

Download versions to compare from GH

Understanding Data Set Diffs

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

Display dataset | Download .csv | Include all modules v | 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

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 `master`

Directory within repository `vendor/cisco/nx/10.3-1`

Include subdirectories

Import YANG files

- 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. On the left is a navigation menu with categories like Admin, Setup, Operations, Analytics, Mapper, Protocols, gNMI, gRPC telemetry, NETCONF, and RESTCONF. The main area shows the configuration for a NETCONF operation. At the top, the breadcrumb is 'YANG Suite / NETCONF / YANG set "IOS XE Switching - Cisco Native" / Modules'. The 'YANG SUITE' logo is on the left, and 'admin' with user icons is on the right. The configuration fields are: 'YANG Set' set to 'IOS XE Switching - Cisco Native', 'Module(s)' set to 'Cisco-IOS-XE-environment-oper', 'NETCONF Operation' set to 'get', and 'Device' set to 'jcohoe-cat9300'. Below these 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 node for 'Cisco-IOS-XE-environment-oper'.

Example NETCONF GET IETF-Interfaces

IETF-interfaces.YANG GET-config operation

The image displays a NETCONF GUI interface. On the left, a tree view shows the YANG schema for 'ietf-interfaces'. The 'link-up-down-trap-enable' node is highlighted. The right pane shows the 'Run RPC(s)' window with the following details:

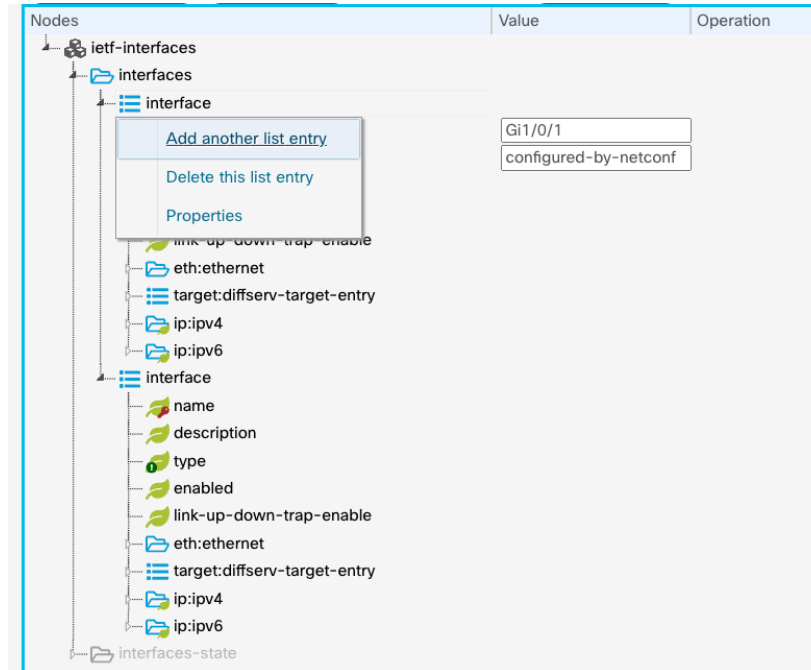
- YANG Set: DevNet Sandbox Modules
- Module(s): iietf-interfaces
- NETCONF Operation: get-config
- Device: sandbox-iosxe-latest-1.cisco.com
- Nodes: iietf-interfaces
- Value: [checked]
- RPC XML:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>
    </filter>
  </get-config>
</rpc>
```


Configure multiple interfaces in a single payload

Get better example (2 ints)

Add xml here 😊



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 displays the YANG Suite interface for a NETCONF operation. The left sidebar shows the YANG Suite navigation menu, including Admin, Setup, Operations, Analytics, Mapper, Protocols, gNMI, gRPC telemetry, NETCONF, RESTCONF, Test Manager, and Help. The main area shows the YANG Set (IOS XE Switching - Cisco Native) and the Module(s) (Cisco-IOS-XE-environment-oper). The NETCONF Operation is set to 'get' for the Device 'jcoho-cat9300'. The YANG Tree shows the 'environment-sensors' node expanded to 'environment-sensor'. The RPC request is shown as:

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

The resulting XML response is shown in the terminal window:

```
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-f1b22a2c9f4">
  <data>
    <environment-sensors xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-environment-oper">
      <environment-sensor>
        <name>Inlet Temp Sena</name>
        <location>Switch 1</location>
        <state>ON</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>
```

Build NETCONF XML Payload to GET Hostname

The screenshot displays the Cisco YANG Suite interface. The top navigation bar includes the title "Cisco YANG Suite" and the current configuration path: "YANG Suite / NETCONF / YANG set 'c9300-default-yangset' / Modules". The user is logged in as "admin".

The main configuration area shows the following settings:

- YANG Set: c9300-default-yangset
- Module(s): Cisco-IOS-XE-native
- NETCONF Operation: get-config
- Device: C9300

Buttons for "Edit Device", "Open Device Window", "Build RPC", "Run RPC(s)", and "Clear RPC(s)" are visible.

The "Nodes" tree on the left shows the hierarchy of the YANG model. The "hostname" node is selected, and its value is "string".

The "Run RPC(s)" panel on the right displays the generated XML payload:

```
<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 interface is divided into several sections:

- Left Sidebar:** A navigation menu with options: Admin, Setup, Analytics, Explore, Protocols, Test Manager, Help.
- Top Bar:** Shows the current context: YANG Suite / NETCONF / YANG set "c9300-default-yangset" / Modules. The user is logged in as 'admin'.
- Configuration Area:**
 - YANG Set:** c9300-default-yangset
 - Module(s):** Cisco-IOS-XE-native
 - NETCONF Operation:** edit-config
 - Device:** C9300
 - Buttons:** Edit Device, Open Device Window, Run RPC(s), Clear RPC(s), Build RPC.
- Nodes Tree:** A hierarchical tree of nodes under 'Cisco-IOS-XE-native'. The 'hostname' node is selected, and its value is set to 'netconf-pro'.
- RPC Payload:** The generated XML payload for the RPC is shown in the right pane:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
  </edit-config>
  <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:base:1.0"
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native"
        <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. The top navigation bar includes 'Admin', 'Setup', 'Explore', 'Protocols', and 'Help'. The main area shows the 'NETCONF' section for the 'c9300-default-yangset' YANG set and the 'Cisco-IOS-XE-native' module. The 'Replays' menu is open, highlighting the 'Generate Python script' option. The 'Nodes' tree on the left shows the configuration structure, with 'ios-vlan:access-log' selected. The right pane shows the generated XML payload for the 'get' operation, and a dark overlay displays the resulting Python script code.

```
<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-config>
</rpc>
```

```
1 #!/usr/bin/env python
2 import lxml.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()
```

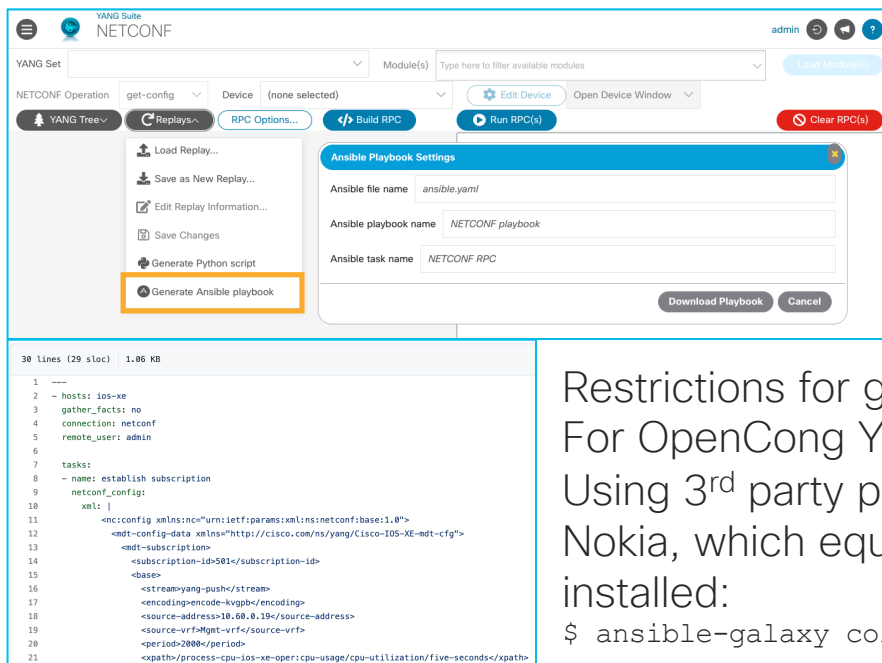

YANG Suite + Ansible

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 NETCONF interface. The left sidebar has the 'NETCONF' option highlighted with an orange box. The main interface shows the 'Generate Ansible playbook' button also highlighted with an orange box. The 'Ansible Playbook Settings' dialog box is open, showing the following fields:

- Ansible file name: ansible.yaml
- Ansible playbook name: NETCONF playbook
- Ansible task name: NETCONF RPC

Below the dialog box, a preview of the generated Ansible playbook is shown:

```
30 Lines (29 sloc) 1.06 KB
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/encoding>
18                    <source-address>10.60.0.19</source-address>
19                    <source-vrf>mgmt-vrf</source-vrf>
20                  <period>300</period>
21                <path>/process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds</path>
```

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 menu with a hamburger icon and a globe icon, followed by the text 'YANG Suite' and '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...' (in a rounded rectangle). 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 red arrow points to the 'Generate Ansible playbook' option.

NETCONF + Ansible Change Host Name demo

The screenshot shows the Cisco YANG Suite web interface. The top navigation bar includes the 'Cisco YANG Suite' logo, a hamburger menu, a 'Help' link, and a 'Welcome to Cisco YANG Suite!' message. 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`.

The right sidebar is titled 'YANG Suite Documentation' and contains a search box 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 'Welcome to Cisco YANG Suite!' message, a brief description of the suite's purpose, and a 'Docker-based Installation' section with a list of steps. The right sidebar contains 'YANG Suite Documentation' with a search bar and a detailed table of contents.

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

Search docs

- 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 XPath
- 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
 - Generating YANG model Tests
 - Managing and editing replay files



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

- [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](#)

Demo: Model Driven Telemetry - NETCONF

Receive CPU telemetry every 10 seconds from NETCONF yang-push

The screenshot shows the NETCONF GUI interface. The top navigation bar includes 'admin' and a help icon. Below it, the 'YANG Set' is 'C9300' and the 'Module(s)' are 'Cisco-IOS-XE-interfaces-oper-x' and 'ietf-event-notifications-x'. The 'NETCONF Operation' is set to '(other RPC)' and the 'Device' is 'C9300'. The 'Build RPC' button is highlighted, and the 'Run RPC(s)' button is also visible. The left pane shows a tree view of the configuration, with 'yp:periodic' selected under 'yp:update-trigger'. The right pane shows the XML configuration for the subscription, including the following code:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="10">
  <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</stream>
    <encoding xmlns:cyp="urn:cisco:params:xml:ns:yang:cisco-xe-ietf-yang-push-ext">cyp.encode-kvg</encoding>
    <xpath-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">/process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds</xpath-filter>
    <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 with the following XML structure:

```
<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-usage>
      </datastore-contents-xml>
    </push-update>
  </notification>
```

The second screenshot shows a received message from the host with the following XML structure:

```
<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

NETCONF and snmpevents streams support subscribe

The NETCONF stream

Configuration changes

The snmpevents stream

SNMP traps

Actions: ✓

- Report device capabilities
- Event notifications (RFC 5277)
 - get streams**
 - subscribe
- Candidate datastore
 - get-config
 - validate
 - commit
 - discard
- Running datastore
 - get-config
- Startup datastore
 - get-config

```
<?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>
```

YANG Suite
NETCONF

YANG Set [dropdown] Module(s) [Type here to filter available modules]

NETCONF Operation [dropdown] Device c9300 [dropdown] Edit Device [button] Open Device Window [dropdown]

YANG Tree [button] Replays [button] RPC Options... [button] Build RPC [button] Run RPC(s) [button]

snmpevents events stream

IOS XE

SNMP

Configure traps (notification) for link down, etc



Connect SNMP feature on IOS XE to the DMI feature

DMI
"NETCONF/YANG"



User:
NETCONF
RESTCONF



Subscribe to
snmpevents



Get SNMP trap notifications:
Link Down notification
Disk full, failed login, etc

Receiver/Tooling

ADD SAMPLE PLZ

NETCONF event stream

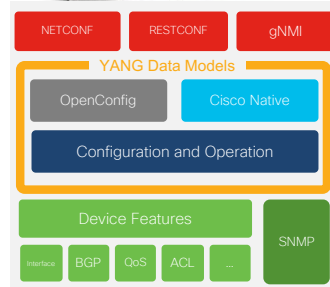
Details of the config change are sent as part of the notification

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



```
jcohoec9300#conf t
Enter configuration commands, one per line. End with CNTL/Z.
jcohoec9300(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</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>
```



- gRPC Telemetry

gRPC Dial-Out with TLS Support

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

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

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 'gRPC Telemetry' are visible. A user profile 'yangsuite-developer' is in the top right.
- Listen at IP address:** A text input field for 'IP address' and a 'Listen at port' section with a 'TCP port' input field. A 'Start receiver' button is present.
- (Optional) TLS receiver:** A checkbox labeled '(Optional) TLS receiver' is highlighted with an orange box. A 'Show receivers' button is next to it.
- TLS Certificates:** 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 has a 'Choose File' button and 'No file chosen' text.
- Telemetry Receivers:** A table with columns 'IP Address', 'Port', and 'TLS'. It shows two entries: one at 10.19.198.133:50070 (TLS: true) and another at 10.19.198.133:57344 (TLS: false). A 'Set Output' button and an 'Exit' button are also present.
- Device Selection:** A dropdown menu 'Choose device with certificate/key' is set to 'ddmi-9500-2'. Below it is a 'Start TLS telemetry receiver' button.
- Data Output:** A 'clear.log' window shows JSON-formatted telemetry data. The data includes fields like 'timestamp', 'subscription', 'node', 'path', and 'fields'. The path is 'Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization'.

Demo: Model Driven Telemetry - gRPC

The screenshot shows the YANG Suite / gRPC Telemetry interface. At the top, it says 'Listen at IP address 127.0.0.1' and 'Listen at port 57344'. There are buttons for 'Stop telemetry receiver' and 'Clear output'. Below this is a table with 6 rows, each representing a subscription configuration. Each row has columns for 'Node', 'Subscription', 'Path', and '/five-seconds'.

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

```
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>

The terminal screenshot shows the following configuration commands and their output:

```
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:

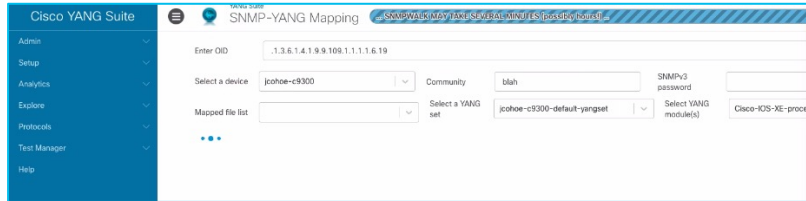
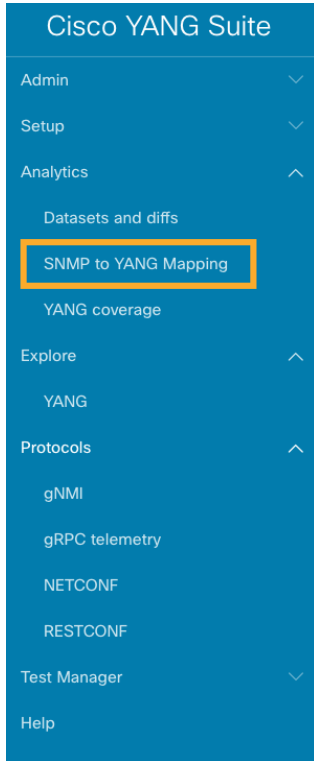
Subscription ID: 202
Type: Configured
State: Valid
Stream: yang-push
Filter:
  Filter type: xpath
  XPath: /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds
Update policy:
  Update Trigger: periodic
  Period: 1000
Encoding: encode-kvgpb
Source VRF:
Source Address:
Notes:

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

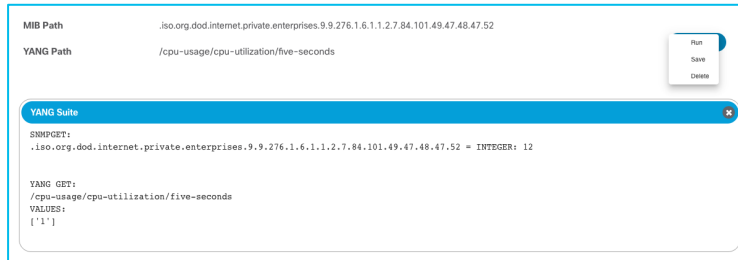
- SNMP OID to YANG XPATH

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.

- 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.

The screenshot shows the YANG Suite RESTCONF interface. At the top left, there is a hamburger menu icon and the text "YANG Suite RESTCONF". Below this, there are four input fields for configuration:

- Select a YANG set:
- Select a device:
- Select YANG module(s):
- Select depth limit:

Below the input fields are three buttons: "Load module(s)", "Generate API(s)", and "Show API(s)". At the bottom, there is a tree view of the YANG model:

- tree icon Cisco-IOS-XE-interfaces-oper
 - folder icon interfaces
 - list icon interface
 - leaf icon name
 - leaf icon interface-type

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

Servers

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

Demo 1: RESTCONF

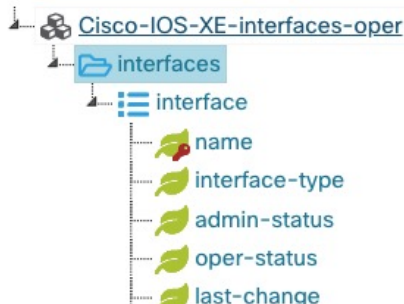
YANG Suite
RESTCONF ready

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 displays the Cisco YANG Suite RESTCONF interface. On the left is a blue navigation sidebar with the following menu items: Admin, Setup, Analytics, Explore, Protocols (expanded to show gNMI, gRPC telemetry, NETCONF, and RESTCONF), Test Manager, and Help. The RESTCONF section is highlighted. The main content area has a header with a hamburger menu, a 'YANG Suite RESTCONF' logo, and a user profile 'admin' with navigation icons. Below the header are four configuration fields: 'Select a YANG set' (c9300-default-yangset), 'Select a device' (C9300), 'Select YANG module(s)' (Cisco-IOS-XE-native), and 'Select depth limit' (2). A 'Load module(s)' button is positioned below the first two fields.

Cisco YANG Suite

YANG Suite RESTCONF

admin

Select a YANG set: c9300-default-yangset

Select a device: C9300

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

Select depth limit: 2

Load module(s)

Demo 2: Acknowledge Notification

The screenshot displays the Cisco YANG Suite RESTCONF interface. On the left is a blue navigation sidebar with the following menu items: Admin, Setup, Analytics, Explore, Protocols (expanded to show gNMI, gRPC telemetry, NETCONF, and RESTCONF), Test Manager, and Help. The RESTCONF section is currently selected. The main content area features a header with the 'YANG Suite RESTCONF' logo and a user profile for 'admin'. Below the header are four selection fields: 'Select a YANG set' (c9300-default-yangset), 'Select a device' (C9300), 'Select YANG module(s)' (Cisco-IOS-XE-native), and 'Select depth limit' (2). Three buttons are visible: 'Load module(s)', 'Generate API(s)', and 'Replays'. A tree view shows 'Cisco-IOS-XE-native' with a tree icon. A blue 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

The screenshot displays the Cisco YANG Suite interface. On the left is a blue sidebar with navigation options: Admin, Setup, Analytics, Explore, Protocols, gNMI, gRPC telemetry, NETCONF, RESTCONF, Test Manager, and Help. The main area contains configuration fields: 'Select a YANG set' (c9300-default-yangset), 'Select YANG module(s)' (Cisco-IOS-XE-native), 'Select a device' (C9300), and 'Select depth limit' (2). Below these are buttons for 'Load module(s)', 'Generate API(s)', and 'Replays'. A tree view on the right shows the hierarchy of YANG modules under 'interface', with 'Loopback' selected at the bottom.

Select a YANG set: c9300-default-yangset

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

Select a device: C9300

Select depth limit: 2

Buttons: Load module(s), Generate API(s), Replays

Tree View:

- 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	▼	🔒

Close

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	
	<small>Controls Accept header.</small>	
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	

Close

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: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: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

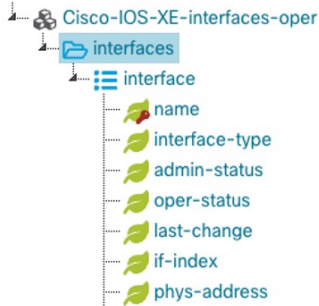
Load module(s)

Generate API(s)

Show API(s)

Generate Python Script

Download Ansible Playbook



What python dependencies/limitations?

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

REST Message Name: interface

Ansi Task Name: interfaces_oper

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 area shows configuration for a RESTCONF operation:

- Select a YANG set:** c9300-default-yangset
- Select a device:** C9300
- Select YANG module(s):** Cisco-IOS-XE-interfaces
- Select depth limit:** No limit

The operation is a **PATCH** to the path `/data/Cisco-IOS-XE-native:native/hostname`. The request body is a JSON object:

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

The interface includes buttons for **Execute** and **Clear**. To the right, a terminal window shows the command output:

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

- gNMI

YANG Suite TLS



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 web interface. At the top, the breadcrumb navigation shows 'YANG Suite / gNMI / YANG set "" / Modules'. The main title is 'gRPC Network Management Interface (gNMI)' with a user 'admin' in the top right corner. Below the title, there are several configuration fields: 'YANG Set' (c9300-default-yangset), 'Module(s)' (Cisco-IOS-XE-interfaces-oper), and 'Device' (C9300). There are buttons for 'Edit Device' and 'Capabilities'. The 'gNMI Operation' section includes buttons for 'Get', 'Set', and 'Subscribe', along with radio buttons for 'All', 'Config', 'State', and 'Operational'. The 'Origin' section has radio buttons for 'Openconfig', 'RFC 7951', and 'Other'. The 'Encoding type' section has radio buttons for 'JSON_IETF' and 'JSON', and a checkbox for 'Prefixing'. Below these are buttons for 'Search XPath...', 'Show Legend', 'Build JSON', and 'Clear Values'. On the right side, there are buttons for 'Run RPC(s)' and 'Clear RPC(s)'. The main content area is divided into two parts: a tree view on the left and a JSON response on the right. The tree view shows a hierarchy starting with 'Cisco-IOS-XE-interfaces-oper', followed by 'interfaces', and then 'interface'. Under 'interface', there are several leaf nodes: 'name', 'interface-type', 'admin-status', 'oper-status', 'last-change', and 'if-index'. The 'name' node is selected, and its value is shown as '*' in a text input field. The JSON response on the right is titled 'gNMI GET Response' and contains the following structure:

```
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

The screenshot displays the YANG Suite gNMI interface for a Cisco IOS-XE device. The interface is titled "gRPC Network Management Interface (gNMI)" and is accessed by the user "admin".

Configuration:

- YANG Set:** c9300-default-yangset
- Module(s):** Cisco-IOS-XE-interfaces-oper
- Device:** C9300
- gNMI Operation:** Get (selected), Set, Subscribe
- Origin:** RFC 7951 (selected), Openconfig, Other
- Encoding type:** JSON_IETF (selected), JSON, Prefixing

Actions: Edit Device, Capabilities, Search XPath(s)..., Show Legend, Build JSON, Clear Values, Run RPC(s), Stop Session

Nodes:

- Cisco-IOS-XE-interfaces-oper
 - interfaces
 - interface
 - name
 - interface-type

Value: [checked]

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\", \"interface-type\": \"iana-iftype\"}]"
    }
  }
  update {
    path {
```

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

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 like Admin, Setup, Analytics, Explore, Protocols, gNMI, gRPC telemetry, NETCONF, RESTCONF, Test Manager, and Help.

Key configuration elements include:

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

The configuration tree on the left shows the following structure:

- 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

The configuration value for the selected 'hostname' node is 'set-by-ys-gnmi'. The 'Operat' column shows 'upda'.

The JSON RPC payload shown in the right pane is:

```
{
  "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 gNMI interface with the following configuration:

- YANG Set:** jcohoe-c9300-default-yangset
- Module(s):** openconfig-system
- Device:** jcohoe-c9300
- gNMI Operation:** Get
- GET type:** All
- Origin:** Openconfig
- Encoding type:** JSON_IETF

The configuration tree on the left shows the path: `openconfig-system > system > config > hostname`. The value for `hostname` is `oc-inet:domain-name`.

The JSON response in the right pane is:

```
{
  "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 shows the Cisco YANG Suite interface for gNMI. The main title is "gRPC Network Management Interface (gNMI)". The interface includes a sidebar on the left with navigation options like Admin, Setup, Analytics, Explore, Protocols, gNMI, gRPC telemetry, NETCONF, RESTCONF, Test Manager, and Help. The main area displays the configuration for the "ietf-interfaces" module on a "C9300" device. The "GET type" is set to "All", the "Origin" is "RFC 7951", and the "Encoding type" is "JSON_IETF". The "Run RPC(s)" button is highlighted in blue, and the "Clear RPC(s)" button is highlighted in red. The right pane shows the JSON response for the GET operation:

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

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	set-by-ys-gnmi	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 Cisco native interfaces

The screenshot shows the gNMI interface with the following configuration:

- YANG Set: jchoe-c9300-default-yangset
- Module(s): Cisco-IOS-XE-interfaces-oper
- Device: jchoe-c9300
- gNMI Operation: Subscribe
- Origin: RFC 7951
- Encoding type: JSON_IETF
- Sample interval: 30
- Buttons: Run RPC(s), Clear Values, Clear RPC(s)

The Nodes tree shows the following structure:

- 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

The JSON output shows the following structure:

```
{
  "subscribe": {
    "subscription": {
      "path": {
        "origin": "rfc7951",
        "elem": [
          {
            "name": "Cisco-IOS-XE-interfaces-oper:interfaces"
          },
          {
            "name": "interface"
          }
        ]
      },
      "mode": "SAMPLE",
      "sampleInterval": "30000000000"
    },
    "encoding": "JSON_IETF"
  }
}
```

The screenshot shows the JSON output for the subscribe operation, which is a list of interface statistics:

```
update {
  timestamp: 2022-07-26T17:43:57.920000+00:00
  update {
    path {
      origin: rfc7951
      elem {
        name: Cisco-IOS-XE-interfaces-oper:interfaces
      }
    }
    phys-address: 70:1f:53:9b:0f:a9
    speed: 1000000000
    statistics {
      discontinuity-time: 2022-07-26T17:40:57+00:00
      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
      out-octets: 4073505065
      out-unicast-pkts: 35917940
      out-broadcast-pkts: 17172777
      out-multicast-pkts: 17949523
      out-discards: 0
      out-errors: 0
      rx-pps: 0
      rx-kbps: 0
      tx-pps: 5
      tx-kbps: 4
      num-flaps: 0
      in-crc-errors: 0
      in-discards-64: 0
      in-errors-64: 0
      in-unknown-protos-64: 0
      out-octets-64: 4073505065
    }
  }
}
```

gNMI Subscribe OpenConfig interfaces

The screenshot shows the gNMI web interface for the 'openconfig-interfaces' module. The 'Subscribe' button is highlighted. The 'Origin' is set to 'Openconfig' and the 'Mode' is 'SAMPLE' with a 'Sample interval' of 30. The 'Encoding type' is 'JSON_IETF'. The 'Nodes' tree on the left shows the hierarchy: openconfig-interfaces > interfaces > interface. The 'Value' column shows checkboxes for 'interfaces' and 'interface'. The 'Operation' column is empty. The main content area displays the JSON RPC payload for the subscription.

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

The screenshot shows the 'JSON Decoded' response for the subscription. The response is a JSON array containing a single object representing the subscribed configuration for the 'FortyGigabitEthernet1/1/1' interface.

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

JSON Decoded
val [
]js {
  {
    "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

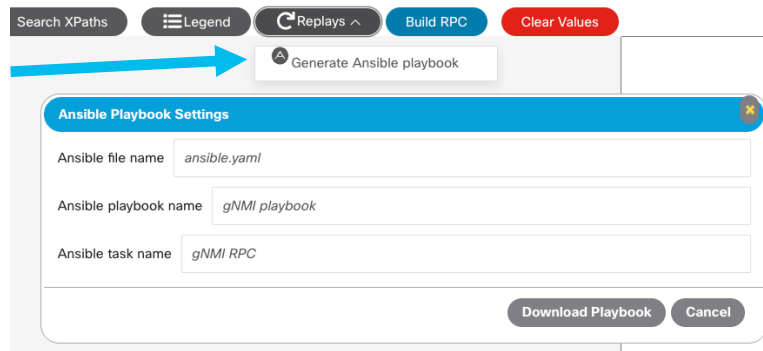
collections:
- nokia.grpc

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

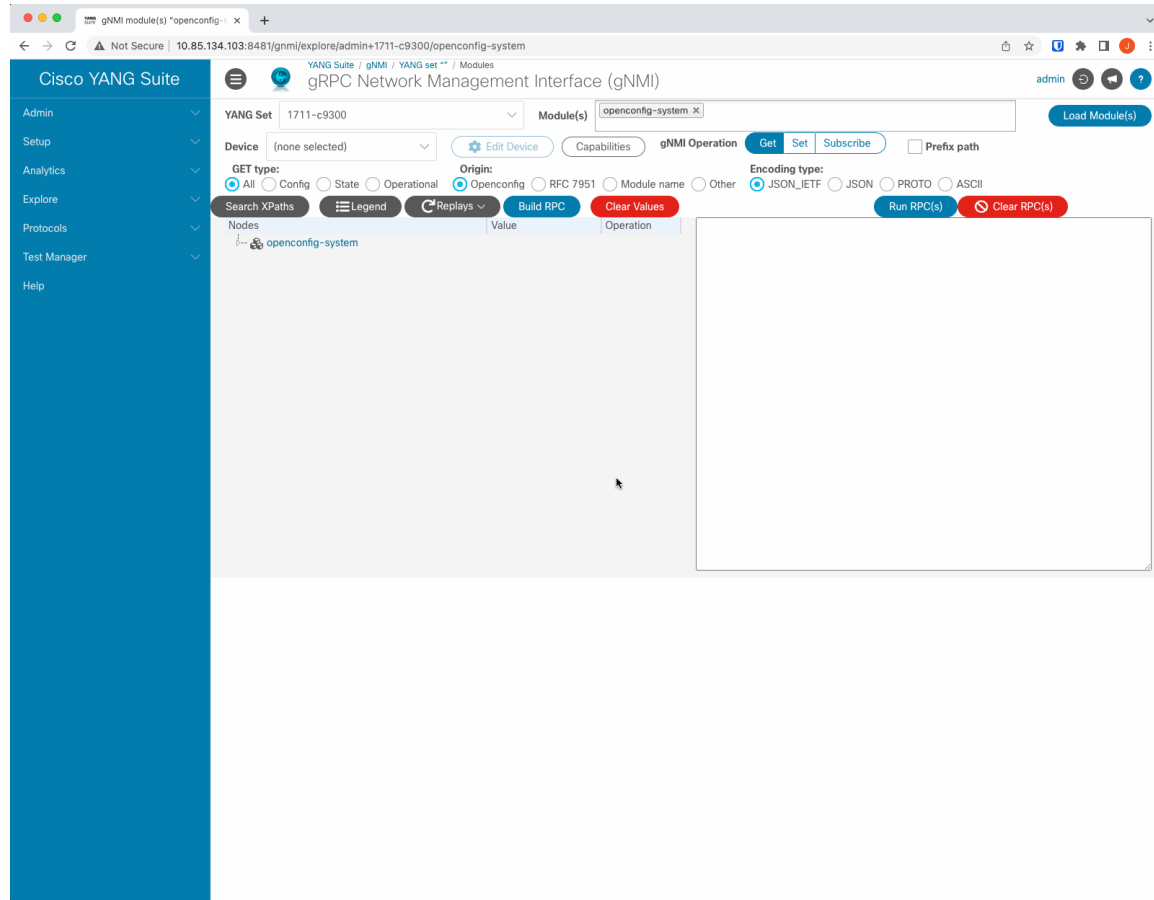
  register: testout
- name: dump test output
  debug:
    msg: '{{ testout.output }}'
```

```
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) web application. The interface is titled "Cisco YANG Suite" and "gRPC Network Management Interface (gNMI)". It features a left-hand navigation menu with options: Admin, Setup, Analytics, Explore, Protocols, Test Manager, and Help. The main content area shows configuration for a "YANG Set" (1711-c9300) and "Module(s)" (openconfig-system). The "Device" is set to "(none selected)". The "GET type" is set to "All". The "Origin" is set to "Openconfig". The "Encoding type" is set to "JSON_IETF". The "Prefix path" checkbox is unchecked. The "Build RPC" button is highlighted in blue, and the "Clear Values" button is highlighted in red. The "Run RPC(s)" and "Clear RPC(s)" buttons are also visible. The interface is currently empty, with no data displayed in the main area.

Additional Tooling:
Postman

Generating Python from POSTMAN

Postman to generate Python

Use Postman to generate Python code with the API call

The screenshot displays the 'GENERATE CODE SNIPPETS' interface in Postman. On the left, a list of programming languages is shown, with 'Python - Requests' selected and highlighted in a blue box. The main area shows the generated Python code for a GET request:

```
1 import requests
2
3 url = "https://10.85.134.119/restconf/data/Cisco-IOS-XE-interfaces-oper:interfaces/interface?fields=name;speed;phys-address;admin-status;statistics/in-octets;statistics/out-octets"
4
5 payload = {}
6 headers = {
7     'Accept': 'application/yang-data+json',
8     'Content-Type': 'application/yang-data+json',
9     'Authorization': 'Basic YWRtaW46Q2lzMzY2Z28xMjM='
10 }
11
12 response = requests.request("GET", url, headers=headers, data = payload)
13
14 print(response.text.encode('utf8'))
15
```

In the top right corner of the code editor, the 'Code' button is highlighted in a blue box. A blue callout box with the text 'Select "Code" from top right and then "Python"' has an arrow pointing to the 'Code' button. Another blue arrow points from the bottom of the callout box to the 'Python - Requests' option in the language list.

Python + RESTCONF

Copy generate code from POSTMAN, add verify=False (SSL)

```
import requests

url = "https://10.85.134.119/restconf/data/Cisco-IOS-XE-interfaces-oper:interfaces/interface?fields=name;speed;phys-address;admin-status;statistics/in-octets;statistics/out-octets"

payload = {}
headers = {
    'Accept': 'application/yang-data+json',
    'Content-Type': 'application/yang-data+json',
    'Authorization': 'Basic YWRtdW46Q2lzY28xMjM='
}

response = requests.request("GET", url, verify=False, headers=headers, data = payload)

print(response.text.encode('utf8'))
```

Example code in Github @

<https://github.com/jeremycohoe/restconf-catalyst-3850>

Run the Python code to get the results

```
jcohoe@jcohoe-ubuntu18-lab:~$ python restconf_c3850.py
/usr/lib/python2.7/dist-packages/urllib3/connectionpool.py:860:
InsecureRequestWarning)
{
  "Cisco-IOS-XE-interfaces-oper:interface": [
    {
      "name": "Vlan1",
      "admin-status": "if-state-down",
      "phys-address": "34:6f:90:f5:88:c7",
      "speed": "1024000000",
      "statistics": {
        "in-octets": "0",
        "out-octets": 0
      }
    },
    {
      "name": "GigabitEthernet0/0",
      "admin-status": "if-state-up",
      "phys-address": "34:6f:90:f5:88:80",
      "speed": "1024000000",
      "statistics": {
        "in-octets": "1353309",
        "out-octets": 13314944
      }
    },
    {
      "name": "GigabitEthernet1/0/1",
      "admin-status": "if-state-down",
      "phys-address": "34:6f:90:f5:88:81",
      "speed": "1024000000",
      "statistics": {
        "in-octets": "0",
        "out-octets": 0
      }
    }
  ]
}
```

RESTCONF + YANGSuite

Resources

YANG Suite Resources

Blogs



March 8, 2021 2 Comments


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

Jeremy Cohoe

This blog is authored by Jeremy Cohoe 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 (ANG) models resulting 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/angallthetime01>



April 26, 2022 Latest Comment

Developer
Snack Minute #58

Developer
YANG Suite updates and new features: A sneak peek

Matt DeHapoll

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 Cohoe (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 Latest Comment

Developer
How Cisco Leverages YANG Suite for YANG Development

Jeremy Cohoe

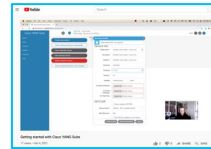
This blog is co-authored by Han Tiku, a Software Engineering Leader with Cisco

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/leverageyangsuite01?dtid=oscscd000283>

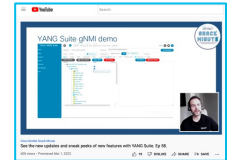
YouTube Videos



<https://youtu.be/smrhIL5Ayz0>

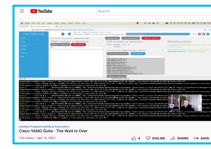


<https://www.youtube.com/watch?v=d1un33611JA>



<https://www.youtube.com/watch?v=soyWPr0fJ0s>

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

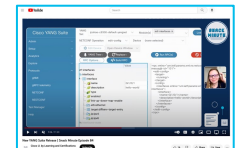


<https://www.youtube.com/watch?v=PkbAOz71vNk>



<https://www.youtube.com/watch?v=3zmNdfn8b38>

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



https://www.youtube.com/watch?v=zVsOO9_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.

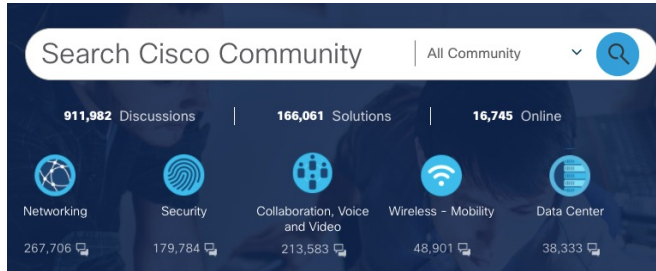


Submit a Github Issue

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/#MaW78CeIS>

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 includes a search bar. A sidebar on the left lists 'USER DOCUMENTATION' with links: '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', featuring a list of links: 'Welcome to Cisco YANG Suite!' (with sub-links for Docker, certificates, access, Python, and installation), 'Managing YANG module files in YANG Suite' (with sub-links for repository and module set), 'Device Profiles' (with sub-link for managing profiles), 'Working with YANG Models' (with sub-link for exploring 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>

Admin

Setup

Analytics

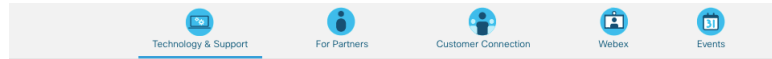
Explore

Protocols

Help

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 IOx 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 cert 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:

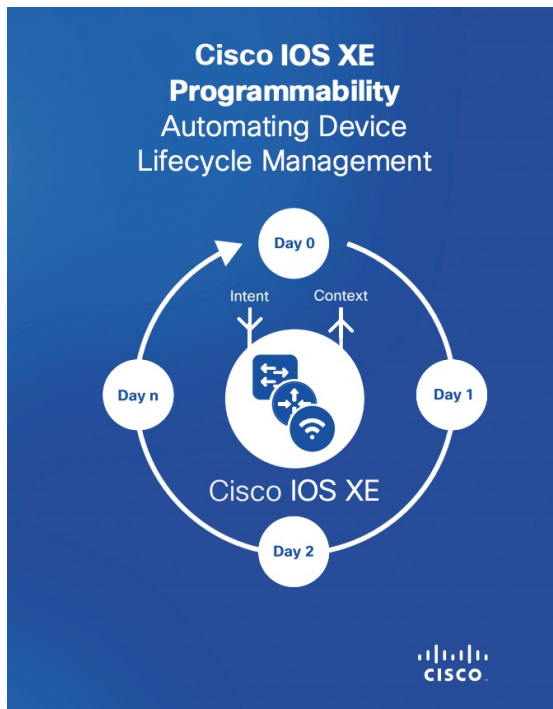
```
docker-compose up
```

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
```

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/179/b_179_programmability_cg.html

Programmability & Automation on SalesConnect

Sales Hubs My Learning Maps My Briefcases More ▾ Cisco SalesConnect

IOS XE Programmability

Page Access Level: Employee | Add to My Briefcase | Share | Subscribe

Programmability & Automation

Overview Products & Solutions Sales Plays Sales Resources Sales Training Launches Help Desk Proposals

IOS XE Programmability & Automation

Programmability and Automation on the IOS XE platform details the programmatic interfaces and YANG data models, the Python API, Model-Driven Telemetry, the Guest Shell Linux container, and a variety of other programmability, automation, and telemetry features are detailed.

Sales Technical

Engage

Sales Presentations (BDM/TDM), Datasheets, FAQs

IOS XE Programmability and Automation TDM
Presentations | Validated: 09 Dec 2021 | Source: Business Unit

Report an Issue Feedback Terms & Conditions Privacy Cookies Trademarks

IOS XE Programmability and Automation TDM P
Presentations | Validated: 09 Dec 2021 | Source: Business Unit

IOS XE Programmability and Automation BDM P
Presentations | Validated: 03 Apr 2020 | Source: Business Unit

YANG Suite TDM P
Presentations | Validated: 23 Feb 2022 | Source: Business Unit

Modernizing IT: Selling & Positioning Programmability and Automation P
Presentations | Validated: 02 Mar 2022 | Source: Business Unit

Terraform with IOS XE P
Presentations | Validated: 22 Apr 2022 | Source: Business Unit

IOS XE Terraform Provider Introduction and Demo A
Other Collateral | Validated: 25 Apr 2022 | Source: Business Unit

Using the YANG Development Kit (YDK) with Cisco IOS XE A
White Papers | Validated: 03 Apr 2020 | Source: Business Unit

Open IOS XE Programmability FAQ A
FAQ | Validated: 01 Apr 2020 | Source: Business Unit

Cisco IOS XE Programmability Book A
Overviews | Validated: 03 Apr 2020 | Source: Business Unit

Ansible TDM P
Presentations | Validated: 01 Apr 2020 | Source: Business Unit

<https://salesconnect.cloudapps.cisco.com/#/program/PAGE-16608>

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 GUI with the following elements highlighted:

- Explorer** tab selected in the top left.
- interface** selected in the left tree view.
- name** selected under the interface tree.
- <get-config>** selected in the 'Values' column.
- Oper** selected in the bottom left 'Operation' dropdown.
- RPC** button highlighted in the 'Operations' section.
- The console area shows the generated NETCONF RPC message:

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running>
    </source>
    <filter>
      <interface xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name>
            </interface>
          </interface>
        </filter>
      </get-config>
    </rpc>
```

Status: Received HTTP result for request type pc

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

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



CISCO

DEVNET

developer.cisco.com



Start Now



Videos and
Tutorials



Sandbox
Learning Lab



Automation and
Code Exchange



Learning and
Certifications



Community and
Study Groups

developer.cisco.com



The bridge to possible