Configure NETCONF/YANG for Cisco IOS XE 16.X Platforms

Contents

Introduction

This document describes how to configure NETCONF/YANG on Cisco IOS® XE 16.x based Platforms.

Prerequisites

Requirements

NETCONF/YANG is supported as of Cisco IOS® XE 16.3.1 software.

Note: No prior experience with NETCONF, YANG, or Python scripting is required in order to use this document.

Components Used

The information in this document is based on these software and hardware versions:

In this example, a stand alone WS-C3850-12X48U switch running Cisco IOS XE 16.3.3 is used as the NETCONF server. This is the device that is configured and from which data (show command output) is collected from via NETCONF/YANG.

A laptop (Apple MacBook Pro running macOS Sierra 10.12.2 and Google Chrome browser) is used as the NETCONF Client. It acts as the centralized management platform and uses the Yang Explorer application. It is the device that creates the YANG formatted requests that are sent to the Catalyst 3850 via NETCONF RPC (Remote Procedure Call) messages to configure and collect data from the Catalyst 3850.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

The example given in this document focuses on lab testing with the Catalyst 3850, however, the information provided also applies to other Cisco IOS XE 16.x platforms such as the Cisco ASR 1000 series routers.

Data Models - Programmatic and Standards Based Configuration and Monitoring

Data models provide an alternate and centralized way to configure Cisco devices (instead of using the Cisco Command Line Interface (CLI) or Simple Network Management Protocol (SNMP)) and to collect operational data (show commands) from Cisco devices. Since the data models are standards based on the same procedure and can be used to configure or collect data from non-Cisco devices as well, it makes them ideal for customers that support multiple vendors. A centralized management platform (for example, a laptop) can be used to configure or collect data from multiple Cisco devices and the data model architecture allows for automating these procedures via Python scripting (two additional key benefits).

Yet Another Next Generation (YANG) Data Modeling Language (RFC 6020)

YANG is a standards based data modeling language used to create device configuration requests or the requests for operational (show command) data. It has a structured format similar to a computer program that is human readable. Several applications are available that can be run on a centralized management platform (for example, a laptop) to create these configuration and operational data requests.

There are both standard (common) YANG data models that apply to all vendors (for example, a request to disable or shut down an ethernet interface can be identical for both Cisco and non-Cisco devices) as well as device (native, vendor specific) data models that facilitate configuring or collecting operational data associated with proprietary vendor features.

Network Configuration (NETCONF) Protocol (RFC 6241)

NETCONF is a standard based and Extensible Markup Language (XML) encoded protocol that provides the transport to communicate the YANG formatted configuration or operational data request from an application that runs on a centralized management platform (for example, a laptop) to the Cisco device that a user wishes to configure or request operational (show command) data from. It provides transaction based services such as aborting the entire configuration request when a portion of that configuration request fails. NETCONF uses a simple Remote Procedure Call (RPF) based mechanism to facilitate communication between a client (centralized management platform script or application) and a server (Cisco switch or router). It uses Secure Shell (SSH) as the transport layer across network devices. Some NETCONF operations include get, get-config, edit-config, and rpc.

Configure

1. Basic Configuration of a Catalyst 3850 Running Cisco XE 16.3.3 Software to Support NETCONF/YANG Data Modeling

```
3850-1# show running-config

netconf-yang -------------------------------------> Enable NETCONF/YANG globally. It may take up to 90

username cisco1 privilege 15 password 0 cisco1 ---> Username/password used for NETCONF-SSH access
```

Note: This is the complete configuration required on the Catalyst 3850 to support NETCONF/YANG Data Modeling but it assumes that no aaa new-model is configured globally (the default) as well. If it is desired to enable AAA (authentication, authorization, and accounting) by configuring aaa new-model, then this configuration is also required at a minimum. You can also expand this to use AAA with a TACACS+ or RADIUS configuration, but this is beyond the scope of this example.

```
aaa new-model

aaa authorization exec default local -------------> Required for NETCONF-SSH connectivity and edit-conf
```

2. Additional (Optional) Configuration to Allow NETCONF/YANG Syslog and SNMP

Event Monitoring

These snmp-server configurations must be present in order to enable the generation of NETCONF notifications (RFC 5277 - Tools 5277) for Syslog messages and for any configured SNMP traps to also generate NETCONF notifications.

Note: While these are the minimum required, additional snmp-server enable entries can be present as well. A client (centralized management platform) registers to receive the NETCONF notification stream from a server (Catalyst 3850) and send a specific subscription RPC (see section 3 of Configuring the Centralized Management Platform (Laptop)).

```
3850-1# show running-config

snmp-server community <string> RW ------------------------------> SNMP gateway in DMI requires communi
netconf-yang cisco-ia snmp-community-string <string> -----------> Configure the same community string 
snmp-server trap link ietf -------------------------------------> enable traps for IETF link up/down
snmp-server enable traps snmp authentication linkdown linkup ---> enable traps for link up/down
snmp-server enable traps syslog --------------------------------> enable traps for Syslog so notificat
snmp-server manager --------------------------------------------> enable snmp-server
```

For Syslog, this configuration must be present for the Data Model Interface (DMI) on the Catalyst 3850 to have the ability to generate NETCONF notifications defined in RFC 5277 when Syslog messages are generated by Ciscod on the Catalyst 3850.

```
logging history debugging -------> required for the generation of any NETCONF notification messages fo
logging snmp-trap emergencies ---> configure 1 or more of the following to control which levels of Sys
logging snmp-trap alerts
logging snmp-trap critical
logging snmp-trap errors
logging snmp-trap warnings
logging snmp-trap notifications
logging snmp-trap informational
logging snmp-trap debugging
```

For SNMP traps, this configuration is required to generate NETCONF notifications. In Cisco XE 16.3.1 software, a maximum of 10 SNMP traps can be configured to generate NETCONF notifications, but this restriction can be removed in a future release. Notification generation for SNMP traps is enabled by default. To disable generating SNMP trap notifications, use this CLI, no netconf-yang cisco-ia snmp-trap-control global-forwarding.

```
netconf-yang cisco-ia snmp-trap-control trap-list 10.3.6.1.6.3.1.1.5.3 --------> LinkDown trap
netconf-yang cisco-ia snmp-trap-control trap-list 10.3.6.1.6.3.1.1.5.4 --------> LinkUp trap
netconf-yang cisco-ia snmp-trap-control trap-list 10.3.6.1.4.1.9.9.41.2.0.1 ---> Syslog generated noti
```

3. Network Connectivity Configuration of the Catalyst 3850 Used in this Example

The Catalyst 3850 management interface GigabitEthernet0/0 is used to connect to the network and to the centralized management platform (a laptop can be used) in this example. Dynamic Host Configuration Protocol (DHCP) has been used to assign IP address 172.16.167.175 to this interface. Alternate configurations can be used on the Catalyst 3850 as long as the laptop can reach the Catalyst 3850 on the Network.

```
3850-1# show running-config

vrf definition Mgmt-vrf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family

interface GigabitEthernet0/0
 vrf forwarding Mgmt-vrf
 ip address dhcp
 negotiation auto

ip route vrf Mgmt-vrf 0.0.0.0 0.0.0.0 172.16.167.161

3850-1# show ip interface brief
```

```
Interface              IP-Address      OK? Method Status Protocol
Vlan1                  10.1.1.1         YES NVRAM  up      up
Vlan10                 10.10.10.1      YES NVRAM  up     up
Vlan20                 10.20.20.1      YES NVRAM  up     up
GigabitEthernet0/0     172.16.167.175  YES DHCP   up     up
Fo1/1/1                unassigned      YES unset  down   down
Fo1/1/2                unassigned      YES unset  down   down
GigabitEthernet1/0/1   unassigned      YES manual up     up
GigabitEthernet1/0/2   unassigned      YES unset  up     up
GigabitEthernet1/0/3   unassigned      YES unset  down   down
GigabitEthernet1/0/4   unassigned      YES unset  down   down
GigabitEthernet1/0/5   unassigned      YES unset  down   down
```

Verify NETCONF/YANG on the Catalyst 3850

1. From the Command Line Interface (CLI) of the Catalyst 3850, this command can be used to ensure that the software processes required to support the Data Model Interface (DMI) on the Catalyst 3850 run once netconf-yang is configured.

```
3850-1# show platform software yang-management process

confd : Running
nesd : Running
syncfd : Running
ncsshd : Running
dmiauthd : Running
vtyserverutild : Running
opdatamgrd : Running
ngnix : Running
```

The next steps are performed from the centralized management platform. In this example, a laptop (Apple MacBook Pro running macOS Sierra 10.12.2) is used that has network access to the Catalyst 3850. The commands are issued from a terminal prompt on the laptop. There is no special application loaded on the laptop at this point.

2. Ensure that the centralized management platform (laptop) can reach the Catalyst 3850 (172.16.167.175) on the network.

<#root>

```
USER1-M-902T:~ USER1$ ping 172.16.167.175

PING 172.16.167.175 (172.16.167.175): 56 data bytes
64 bytes from 172.16.167.175: icmp_seq=0 ttl=247 time=3.912 ms
64 bytes from 172.16.167.175: icmp_seq=1 ttl=247 time=6.917 ms
64 bytes from 172.16.167.175: icmp_seq=2 ttl=247 time=4.063 ms
64 bytes from 172.16.167.175: icmp_seq=3 ttl=247 time=4.371 ms

^C
```

3. Verify SSH connectivity to the Catalyst 3850 (172.16.167.175 in this example) from the centralized management platform (laptop) with the username and password (cisco1/cisco1) from this Catalyst 3850 configuration. The response can be a long list of NETCONF capabilities from the Catalyst 3850 followed by a hello message. TCP port 830 = netconf-ssh.

Tip: If this SSH test does not work, ensure that any firewall in between the laptop and Catalyst 3850 permits TCP port 830 (reference RFC 4742: Tools 4742).

```
USER1-M-902T:~ USER1$ ssh -s cisco1@172.16.167.175 -p 830 netconf
cisco1@172.16.167.175's password: cisco1

<?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:xpath:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.1</capability>
<capability>urn:ietf:params:netconf:capability:rollback-on-error:1.0</capability
--snip--
</capabilities>
<session-id>2870</session-id></ hello>]]>]]>

Use < ^C > to exit
```

Configure the Centralized Management Platform (Laptop)

1. Install the Yang Explorer Application on a Laptop

In this example, the Yang Explorer application is used on a laptop (Apple MacBook Pro running macOS Sierra 10.12.2, Google Chrome browser) to act as the centralized management platform. Yang Explorer allows the user to do this:

- Upload / Compile YANG data models from User Interface or Command Line
- Build NETCONF RPCs (Remote Procedure Calls)
- Execute RPC against a real NETCONF server (Catalyst 3850)
- Save created RPCs to collections for later use
- Browse data model trees and inspect YANG properties

Note: The YANG Explore application is also supported on Linux systems.

2. Use the Yang Explorer Application

Start the Yang Explorer Application - from a terminal prompt on the laptop run the **./start.sh** and command from the yang-explorer directory.

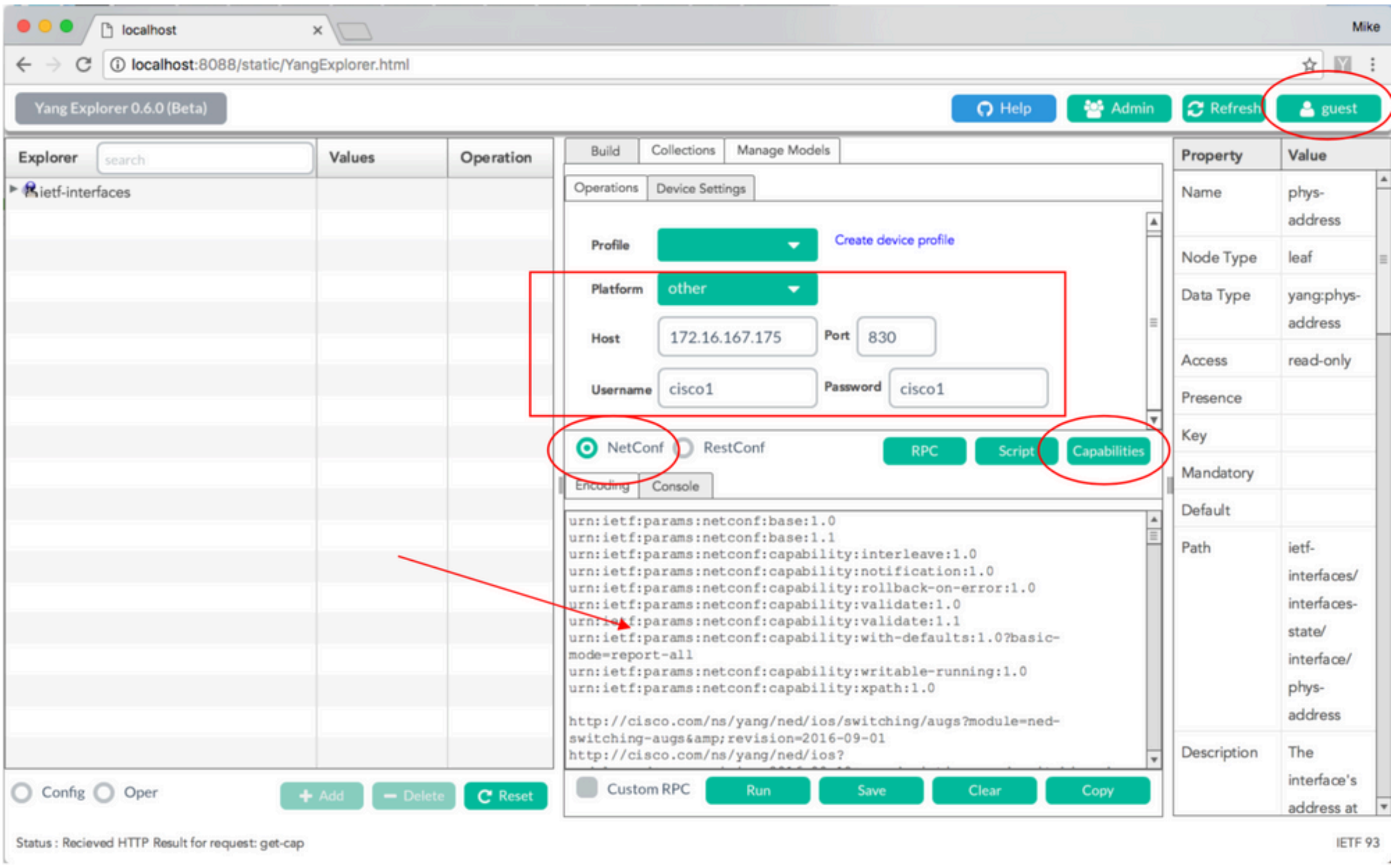
Note: Keep this terminal session open, otherwise, the Yang Explorer application can shut down and must be restarted. It can also serve as a console log of application activity.

```
USER1-M-902T:~ USER1$ cd yang-explorer

USER1-M-902T:yang-explorer USER1$ ./start.sh &

 Starting YangExplorer server ..
 Use http://localhost:8088/static/YangExplorer.html

Performing system checks...

System check identified no issues (0 silenced).
 January 19, 2017 - 23:12:20
 Django version 1.8.3, using settings 'server.settings'
 Starting development server at http://localhost:8088/
 Quit the server with CONTROL-C.
```

Launch the Yang Explorer GUI - Launch the Yang Explorer application GUI and log in to the Yang Explorer application GUI as guest/guest in the top right corner of the application GUI main menu (refer to the screencapture).

Retrieve capabilities from the Catalyst 3850. Enter the Catalyst 3850 details (IP address, Username/Password, TCP port 830 for ssh-netconf) and click **Capabilities** to retrieve the YANG operational capabilities list from the Catalyst 3850 software.

Tip: This is also a good test to confirm that NETCONF communication works between the Yang Explorer application on the Centralized Management Platform (Laptop) and the Catalyst 3850.



Profile

Platform: other

Host: 172.16.167.175 Port: 830

Username: cisco1 Password: cisco1

● NetConf ○ RestConf

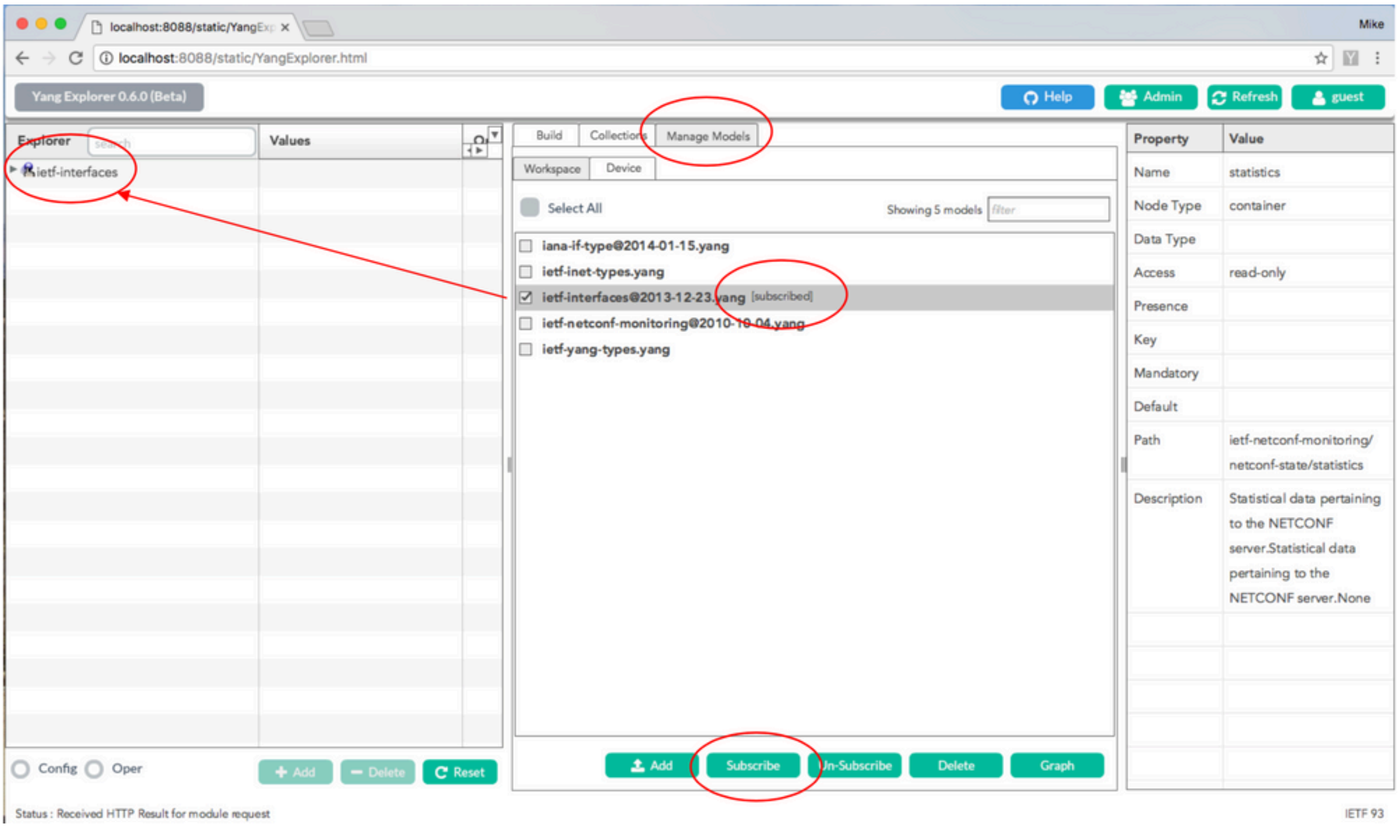
```
urn:ietf:params:netconf:base:1.0
urn:ietf:params:netconf:base:1.1
urn:ietf:params:netconf:capability:interleave:1.0
urn:ietf:params:netconf:capability:notification:1.0
urn:ietf:params:netconf:capability:rollback-on-error:1.0
urn:ietf:params:netconf:capability:validate:1.0
urn:ietf:params:netconf:capability:validate:1.1
urn:ietf:params:netconf:capability:with-defaults:1.0?basic-
mode=report-all
urn:ietf:params:netconf:capability:writable-running:1.0
urn:ietf:params:netconf:capability:xpath:1.0

http://cisco.com/ns/yang/ned/ios/switching/augs?module=ned-
switching-augs&amp;revision=2016-09-01
http://cisco.com/ns/yang/ned/ios?
```

Property	Value
Name	phys-address
Node Type	leaf
Data Type	yang:phys-address
Access	read-only
Presence	
Key	
Mandatory	
Default	
Path	ietf-interfaces/interfaces-state/interface/phys-address
Description	The interface's address at

Status : Recieved HTTP Result for request: get-cap

Load Yang Data Models - Various YANG data models can be subscribed to under Manage Models. Once subscribed, they appear in the Explorer box on the left. These YANG models allow the Yang Explorer application to create YANG formatted NETCONF Remote Procedure Calls (RPC) messages (which are sent to the Catalyst 3850 to configure it or retrieve data from it) without the need to have in depth YANG expertise. Examples of how to do this are covered in the next section Basic NETCONF/YANG Operational

Examples:



Select All — Showing 5 models

- □ iana-if-type@2014-01-15.yang
- □ ietf-inet-types.yang
- ☑ ietf-interfaces@2013-12-23.yang [subscribed]
- □ ietf-netconf-monitoring@2010-10-04.yang
- □ ietf-yang-types.yang

Property	Value
Name	statistics
Node Type	container
Data Type	
Access	read-only
Presence	
Key	
Mandatory	
Default	
Path	ietf-netconf-monitoring/ netconf-state/statistics
Description	Statistical data pertaining to the NETCONF server.Statistical data pertaining to the NETCONF server.None

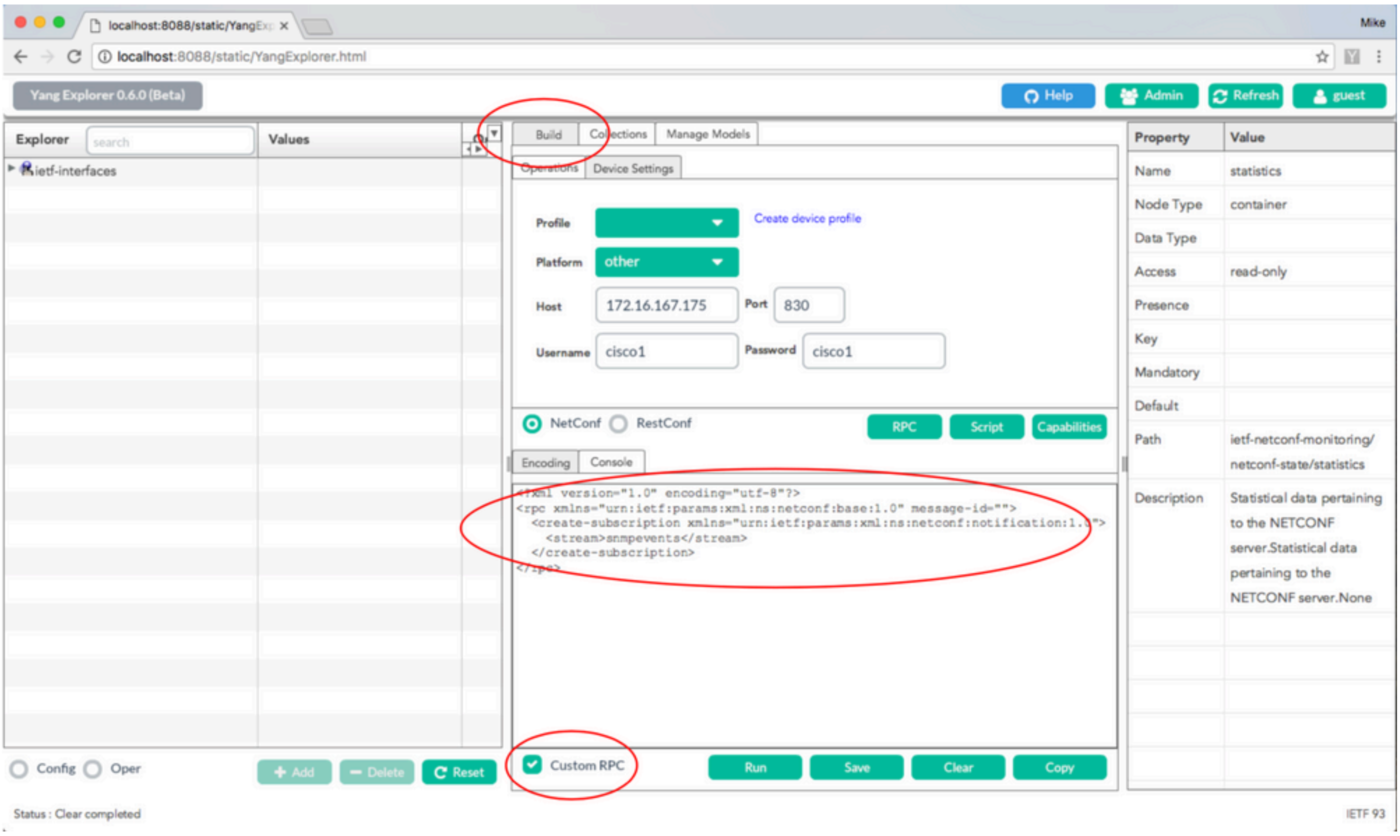
Status : Received HTTP Result for module request

3. Subscribe to NETCONF Notifications (Optional)

A client (centralized management platform) registers to receive NETCONF notification streams from a server (Catalyst 3850) by sending this YANG formatted NETCONF RPC message. The Catalyst 3850 sends NETCONF notifications asynchronously to each client that subscribes. Before you complete this task, ensure that the correct configuration is in place on the Catalyst 3850 to support NETCONF Notifications (see section 2) of Configuring NETCONF/YANG on the Catalyst 3850. The NETCONF server (Catalyst 3850) begins to send the event notifications to the NETCONF client (Centralized Management Platform) as the events occur within the system. These event notifications can continue to be sent until either the NETCONF session is terminated or the subscription terminates for some other reason. See RFC 5277 for more details related to subscription options Tools 5277.

```
<?xml version="1.0" encoding="utf-8"?>
 <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
   <create-subscription xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
     <stream>snmpevents</stream>
   </create-subscription>
 </rpc>
```

To do this, you need to cut and paste this into the Yang Explorer application GUI as a Custom RPC.



Profile
Platform: other
Host: 172.16.167.175 Port: 830
Username: cisco1 Password: cisco1

● NetConf ○ RestConf

```
<?xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
  <create-subscription xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <stream>snmpevents</stream>
  </create-subscription>
</rpc>
```

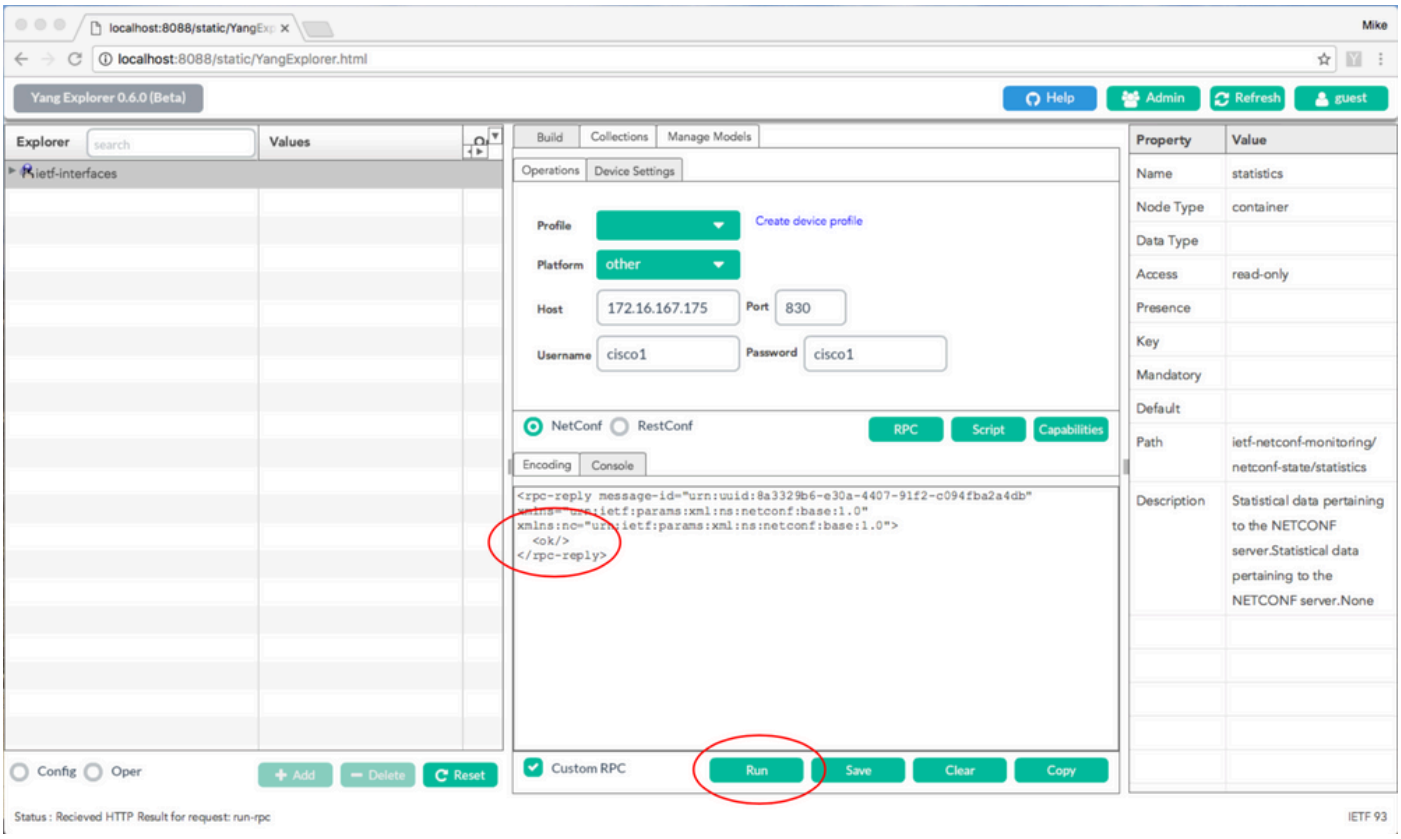
☑ Custom RPC

○ Config ○ Oper

Property	Value
Name	statistics
Node Type	container
Data Type	
Access	read-only
Presence	
Key	
Mandatory	
Default	
Path	ietf-netconf-monitoring/ netconf-state/statistics
Description	Statistical data pertaining to the NETCONF server.Statistical data pertaining to the NETCONF server.None

Status : Clear completed

Next, **Run** is selected in order to send the custom RPC message to the Catalyst 3850 via NETCONF. The Catalyst 3850 replies back with an ok message to let the user know that the operation was successful.



Profile
Platform: other
Host: 172.16.167.175 Port: 830
Username: cisco1 Password: cisco1

● NetConf ○ RestConf

```
<rpc-reply message-id="urn:uuid:8a3329b6-e30a-4407-91f2-c094fba2a4db"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

☑ Custom RPC

○ Config ○ Oper

Property	Value
Name	statistics
Node Type	container
Data Type	
Access	read-only
Presence	
Key	
Mandatory	
Default	
Path	ietf-netconf-monitoring/ netconf-state/statistics
Description	Statistical data pertaining to the NETCONF server.Statistical data pertaining to the NETCONF server.None

Status : Recieved HTTP Result for request: run-rpc

Note: The current version of Yang Explorer used in this example does not have an option to look at the received NETCONF Notifications. They are typically stored in a clickable Notification log in the main menu of the application.

Basic NETCONF/YANG Operational Examples

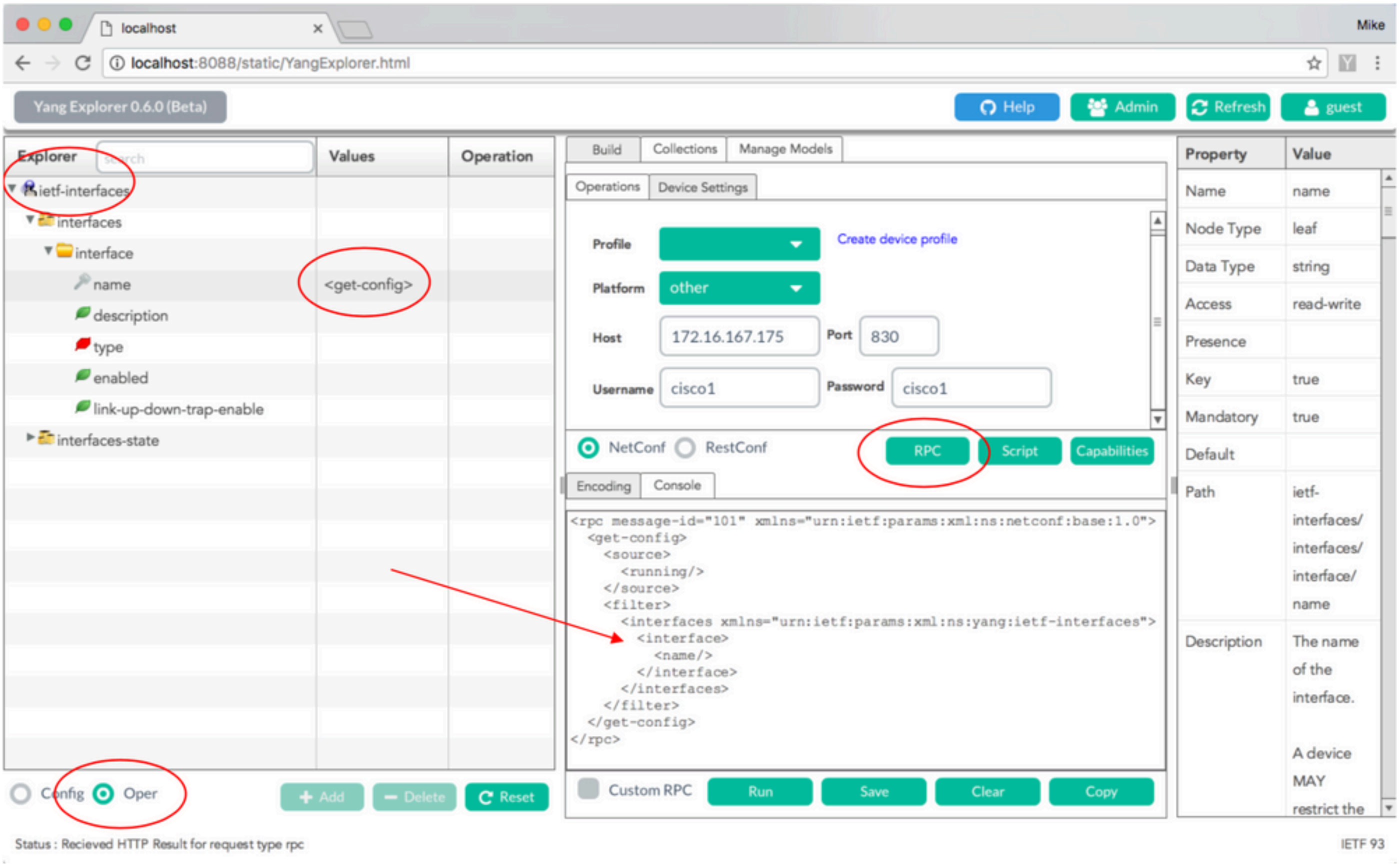
Now that the Catalyst 3850 and the Centralized Management Platform are configured and have started to communicate, lets us look at some basic operational examples.

The examples can demonstrate that the YANG formatted NETCONF RPC messages sent via NETCONF from the Centralized Management Platform (Laptop) Yang Explorer application to the Catalyst 3850 are converted to standard Cisco IOS CLI by the confd software process on the Catalyst 3850. Also, Cisco IOS CLI data (show command data) is converted to YANG formatted data by the confd software process on the Catalyst 3850 before it is sent as NETCONF RPC message to the Centralized Management Platform (Laptop) Yang Explorer application. This means that the regular CLI can still be used on the Catalyst 3850 to configure the switch and collect show command data in addition to use NETCONF/YANG to do the same.

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.



```
▾ ietf-interfaces
  ▾ interfaces
    ▾ interface
        name                      <get-config>
        description
        type
        enabled
        link-up-down-trap-enable
  ▸ interfaces-state
```

○ Config ● Oper

Profile:
Platform: other
Host: 172.16.167.175 Port: 830
Username: cisco1 Password: cisco1

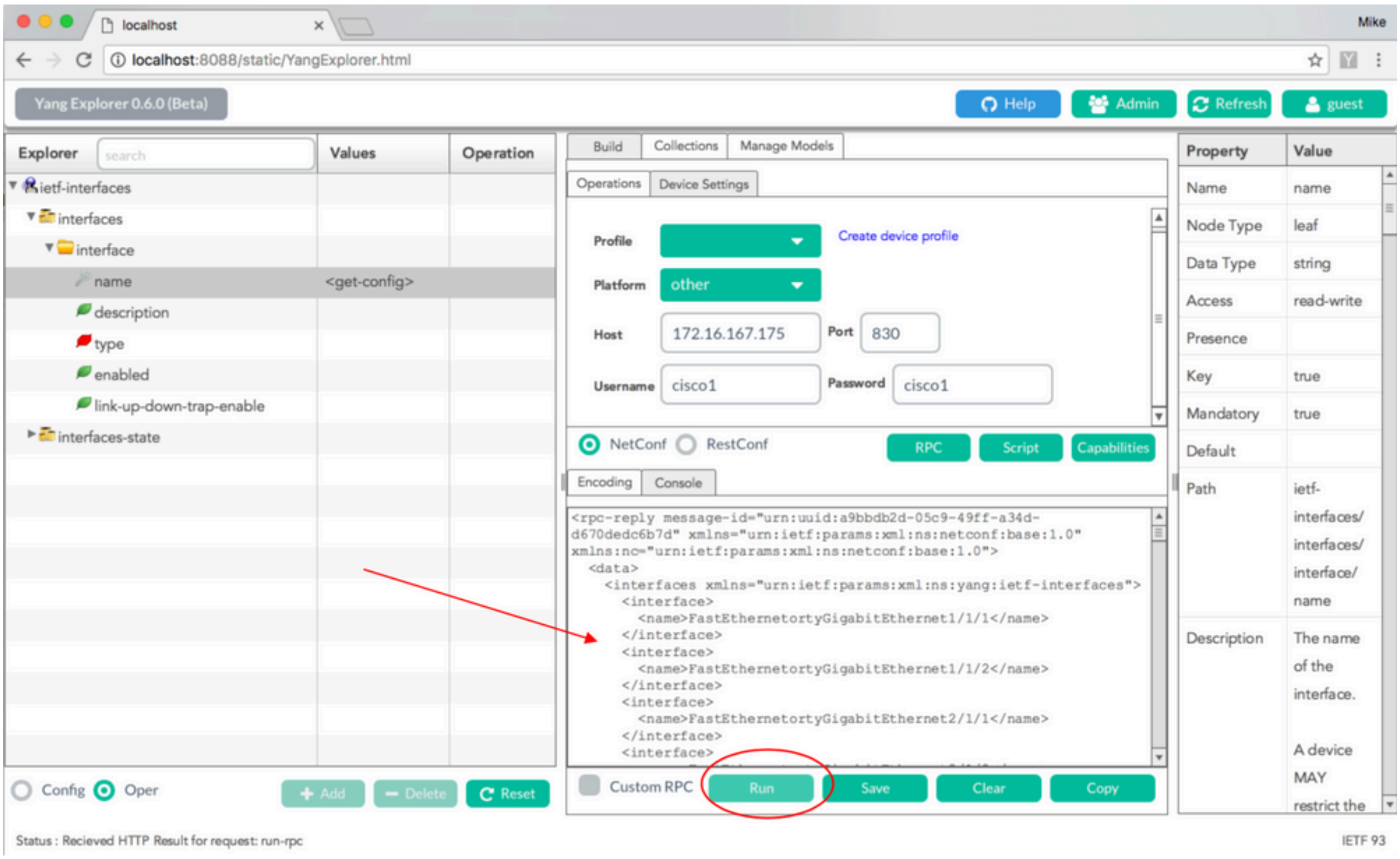
● NetConf ○ RestConf

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

Property	Value
Name	name
Node Type	leaf
Data Type	string
Access	read-write
Presence	
Key	true
Mandatory	true
Default	
Path	ietf-interfaces/interfaces/interface/name
Description	The name of the interface. A device MAY restrict the

Status : Recieved HTTP Result for request type rpc

After the YANG formatted NETCONF RPC message is generated, **Run** is selected in order to send it to the Catalyst 3850. The Catalyst 3850 replies with a YANG formatted (human readable) list of the Catalyst 3850 interface names (GigabitEthernet1/1/1, GigabitEthernet1/1/2, and so on).



Profile
Platform other
Host 172.16.167.175 Port 830
Username cisco1 Password cisco1

```
<rpc-reply message-id="urn:uuid:a9bbdb2d-05c9-49ff-a34d-
d670dedc6b7d" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
      <interface>
        <name>FastEthernetortyGigabitEthernet1/1/1</name>
      </interface>
      <interface>
        <name>FastEthernetortyGigabitEthernet1/1/2</name>
      </interface>
      <interface>
        <name>FastEthernetortyGigabitEthernet2/1/1</name>
      </interface>
      <interface>
```

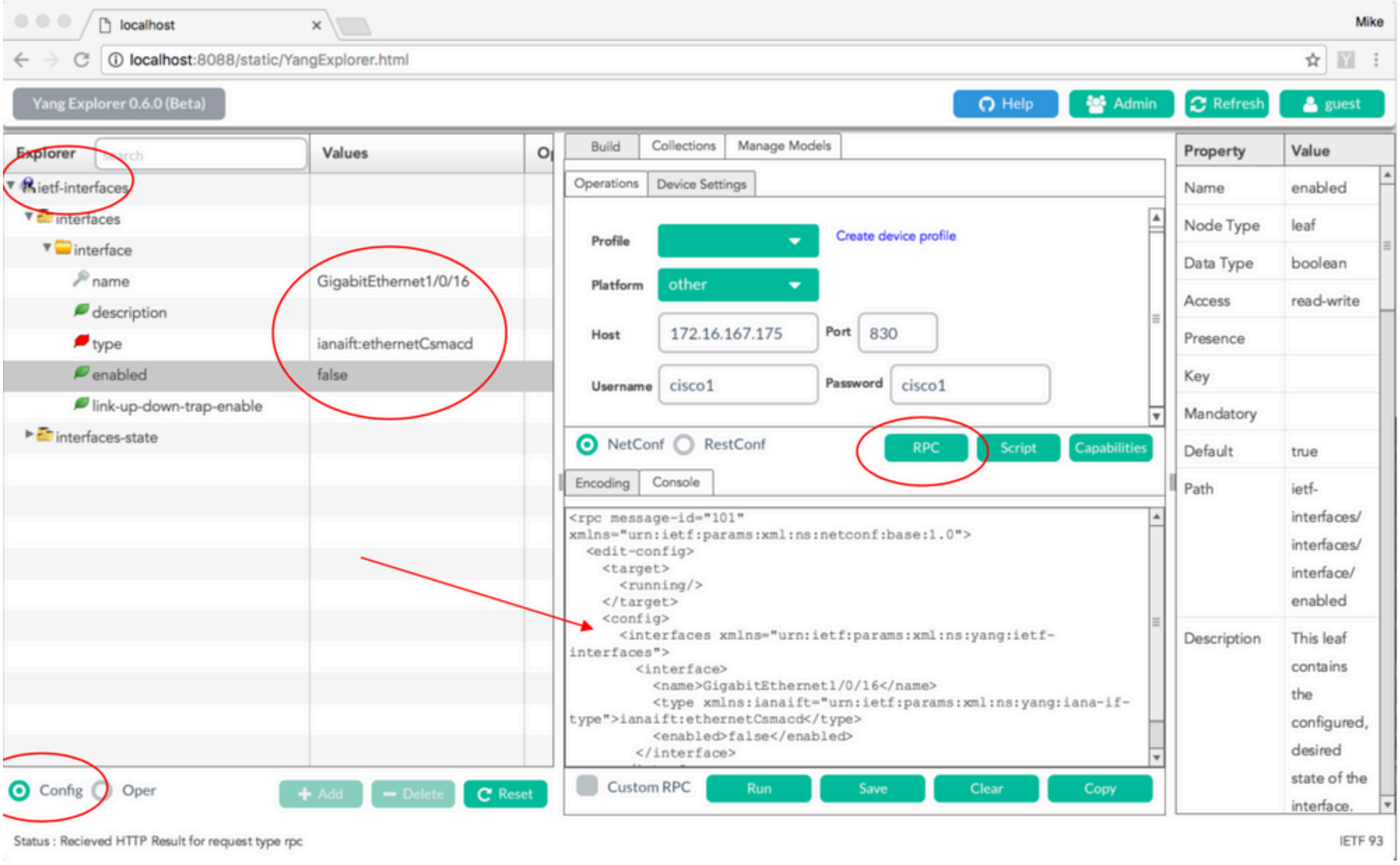
Property	Value
Name	name
Node Type	leaf
Data Type	string
Access	read-write
Presence	
Key	true
Mandatory	true
Default	
Path	ietf-interfaces/interfaces/interface/name
Description	The name of the interface. A device MAY restrict the

Status : Recieved HTTP Result for request: run-rpc

2. Configuration Example

Shut Down an Ethernet Interface on the Catalyst 3850

The desired operation is selected from the left side of the Explorer section of the Yang Explorer application GUI. In this case, to configure an interface (shutting down an interface) is required on the Catalyst 3850 and so **Config** (for configuration) is selected, followed by the required operational parameters under the interface drop down menus. **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 execute the configuration task.



Explorer	Values
ietf-interfaces	
interfaces	
interface	
name	GigabitEthernet1/0/16
description	
type	ianaift:ethernetCsmacd
enabled	false
link-up-down-trap-enable	
interfaces-state	

● Config ○ Oper

Profile:
Platform: other
Host: 172.16.167.175 Port: 830
Username: cisco1 Password: cisco1

● NetConf ○ RestConf

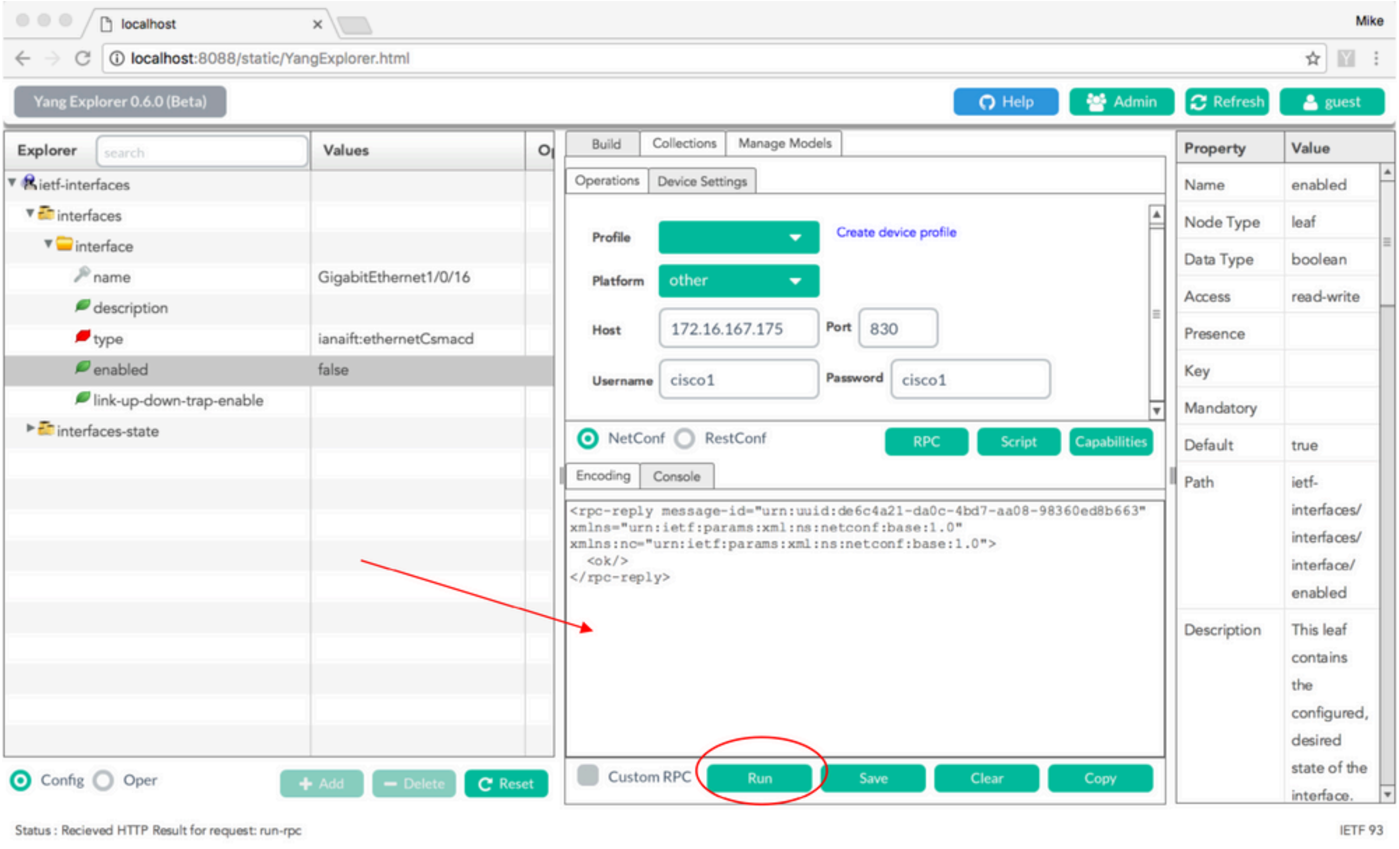
```
<rpc message-id="101"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-
interfaces">
        <interface>
          <name>GigabitEthernet1/0/16</name>
          <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-
type">ianaift:ethernetCsmacd</type>
          <enabled>false</enabled>
        </interface>
```

□ Custom RPC

Property	Value
Name	enabled
Node Type	leaf
Data Type	boolean
Access	read-write
Presence	
Key	
Mandatory	
Default	true
Path	ietf-interfaces/interfaces/interface/enabled
Description	This leaf contains the configured, desired state of the interface.

Status : Recieved HTTP Result for request type rpc

After the YANG formatted NETCONF RPC message is generated, **Run** is selected in order to send it to the Catalyst 3850. The Catalyst 3850 replies with a YANG formatted (human readable) message that state that the configuration operation was successful (ok).



Explorer	Values
ietf-interfaces	
interfaces	
interface	
name	GigabitEthernet1/0/16
description	
type	ianaift:ethernetCsmacd
enabled	false
link-up-down-trap-enable	
interfaces-state	

● Config ○ Oper

Profile:
Platform: other
Host: 172.16.167.175 Port: 830
Username: cisco1 Password: cisco1

● NetConf ○ RestConf

```
<rpc-reply message-id="urn:uuid:de6c4a21-da0c-4bd7-aa08-98360ed8b663"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

□ Custom RPC

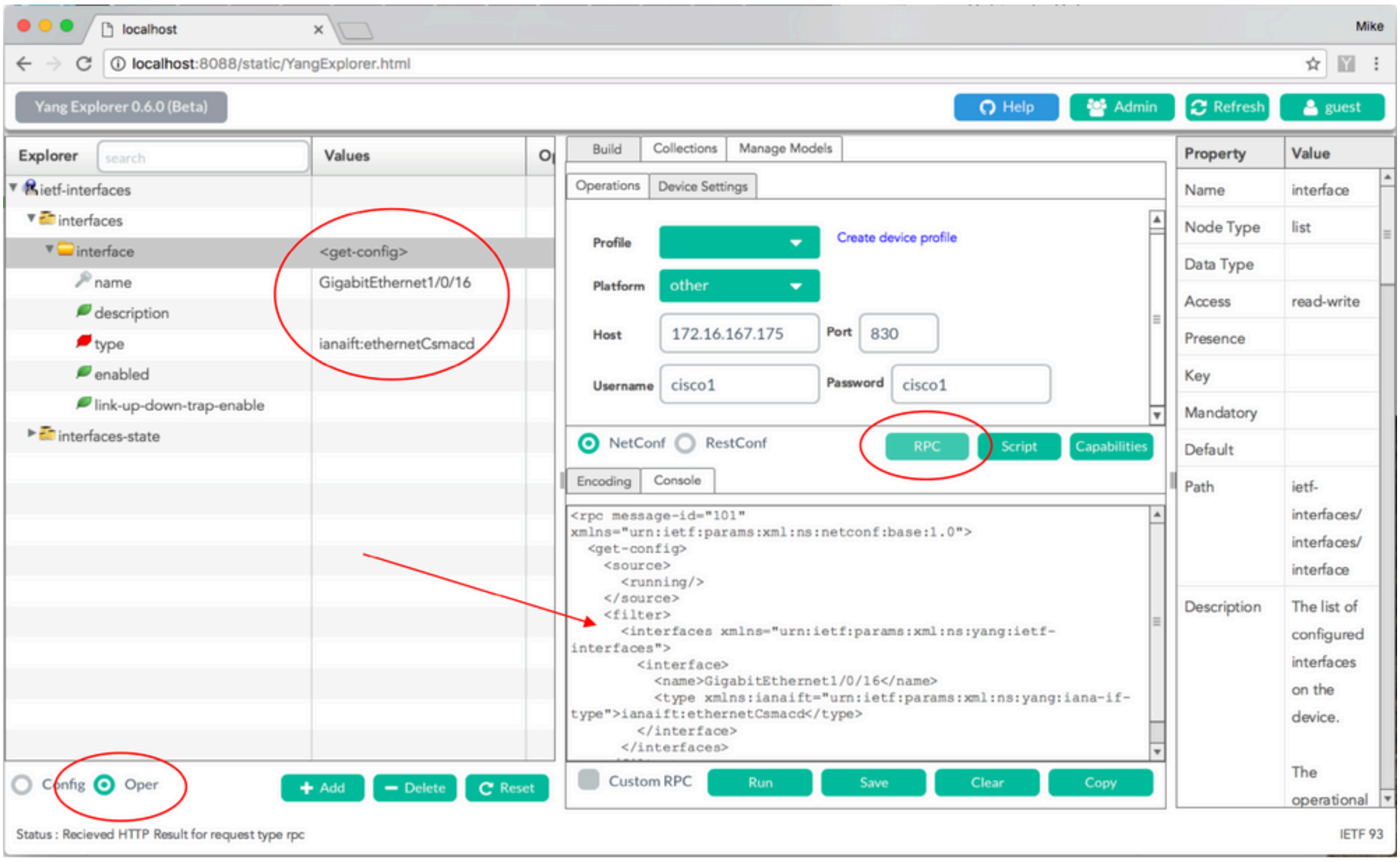
Property	Value
Name	enabled
Node Type	leaf
Data Type	boolean
Access	read-write
Presence	
Key	
Mandatory	
Default	true
Path	ietf-interfaces/interfaces/interface/enabled
Description	This leaf contains the configured, desired state of the interface.

Status : Recieved HTTP Result for request: run-rpc

In order to confirm that the change took place, the configuration can be checked. A get-config operation

(Oper) can be used where the Catalyst 3850 replies back that the interface GigabitEthernet 1/0/16 configuration has enabled = false now which means that the interface was shut down.

Tip: In general, when it is not clear what format the Values can be in the Explorer section of the Yang Explorer application, dumping the YANG formatted Catalyst 3850 configuration, as shown, is a good way to determine what they are before an attempt is made to modify them. The right hand side of the next screens provide some descriptions and dependencies for these values as well in the Property and Value columns.



Explorer	Values
ietf-interfaces	
interfaces	
interface	<get-config>
name	GigabitEthernet1/0/16
description	
type	ianaift:ethernetCsmacd
enabled	
link-up-down-trap-enable	
interfaces-state	

○ Config ● Oper

Profile:

Platform: other

Host: 172.16.167.175 Port: 830

Username: cisco1 Password: cisco1

● NetConf ○ RestConf

```
<rpc message-id="101"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-
interfaces">
        <interface>
          <name>GigabitEthernet1/0/16</name>
          <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-
type">ianaift:ethernetCsmacd</type>
        </interface>
      </interfaces>
```

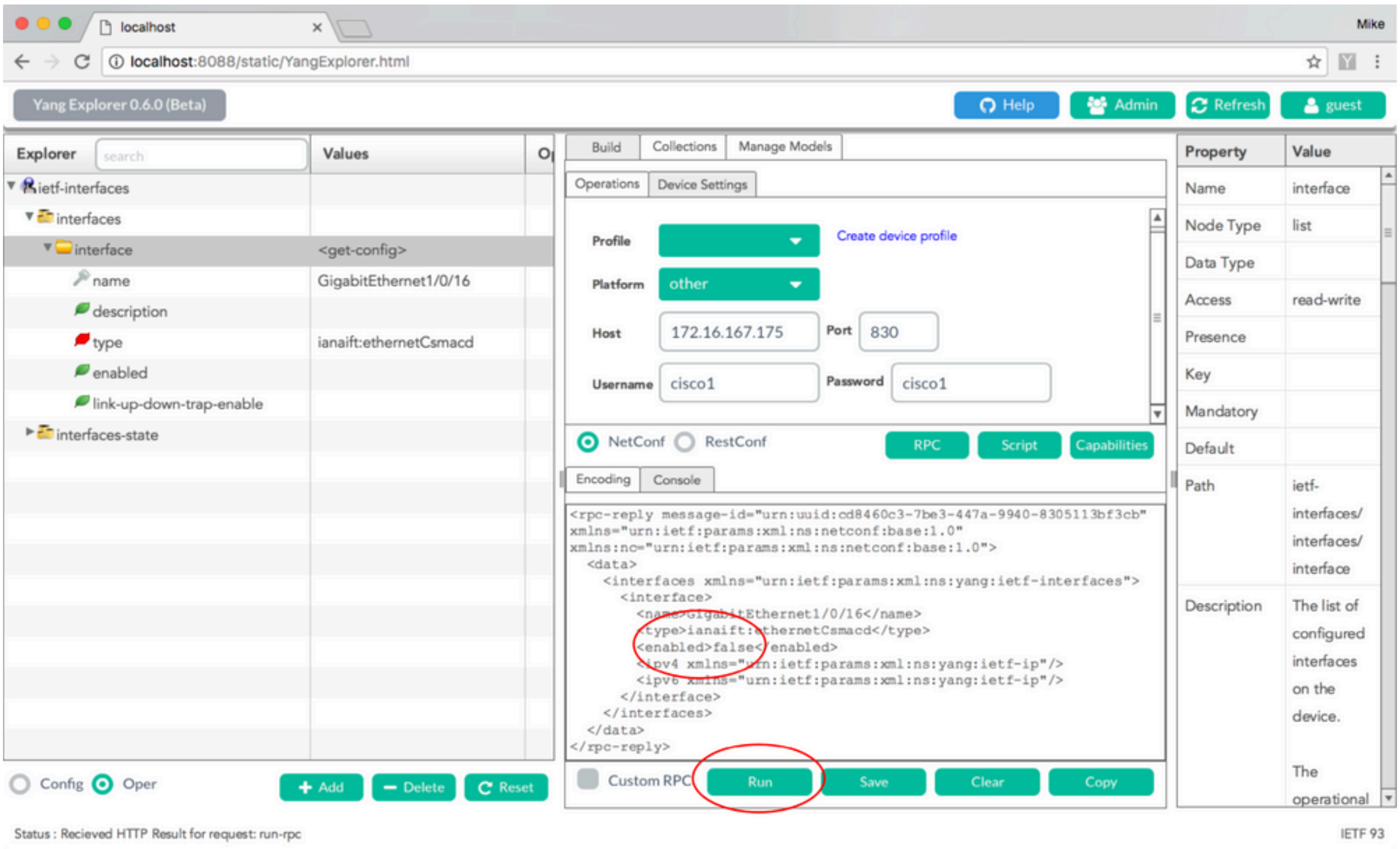
□ Custom RPC

Property	Value
Name	interface
Node Type	list
Data Type	
Access	read-write
Presence	
Key	
Mandatory	
Default	
Path	ietf-interfaces/interfaces/interface
Description	The list of configured interfaces on the device. The operational

Status : Recieved HTTP Result for request type rpc

IETF 93

After the YANG formatted NETCONF RPC message is generated, **Run** is selected in order to send it to the Catalyst 3850. The Catalyst 3850 replies with a YANG formatted message that states that the interface GigabitEthernet 1/0/16 configuration has enabled = false now which means that the interface was shut down.



Explorer	Values
ietf-interfaces	
interfaces	
interface	<get-config>
name	GigabitEthernet1/0/16
description	
type	ianaift:ethernetCsmacd
enabled	
link-up-down-trap-enable	
interfaces-state	

○ Config ● Oper

Profile:

Platform: other

Host: 172.16.167.175 Port: 830

Username: cisco1 Password: cisco1

● NetConf ○ RestConf

```
<rpc-reply message-id="urn:uuid:cd8460c3-7be3-447a-9940-8305113bf3cb"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
      <interface>
        <name>GigabitEthernet1/0/16</name>
        <type>ianaift:ethernetCsmacd</type>
        <enabled>false</enabled>
        <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"/>
        <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"/>
      </interface>
    </interfaces>
  </data>
</rpc-reply>
```

□ Custom RPC

Property	Value
Name	interface
Node Type	list
Data Type	
Access	read-write
Presence	
Key	
Mandatory	
Default	
Path	ietf-interfaces/interfaces/interface
Description	The list of configured interfaces on the device. The operational

Status : Recieved HTTP Result for request: run-rpc

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

At the time of the previous Yang Explorer configuration change operation, this is output from the CLI of the Catalyst 3850. Interface GigabitEthernet 1/0/16 was in the default no shutdown state until the NETCONF RPC message is received as seen in the log message on the Catalyst 3850. After the NETCONF RPC message is received that contains the YANG formatted request to shutdown the interface, the operation is completed, the interface is shutdown, and the running configuration is modified to reflect this. This also demonstrates how the confd software process on the Catalyst 3850 converts the received YANG formatted NETCONF RPC message into standard Cisco IOS CLI. This means that a user can still use regular Cisco IOS CLI to modify the configuration and execute show commands in addition to using NETCONF/YANG to do the same.

```
3850-1# show running-config interface gigabitEthernet 1/0/16 □
Building configuration...

Current configuration : 39 bytes □
! □
interface GigabitEthernet1/0/16
□end

3850-1# show startup-config | begin 1/0/16
□interface GigabitEthernet1/0/16 □
!

*Jan 5 17:05:55.345: %DMI-5-CONFIG_I:Switch 1 R0/0: nesd: Configured from NETCONF/RESTCONF by cisco1, t
*Jan 5 17:05:57.335: %LINK-5-CHANGED: Interface GigabitEthernet1/0/16, changed state to administrativel

3850-1# show running-config interface gigabitEthernet 1/0/16
□Building configuration...
```

```
Current configuration : 49 bytes □
!
□interface GigabitEthernet1/0/16 □
shutdown -------------------------> the interface is shutdown now
□end

3850-1#
```

Note: The configuration has not been saved (copied from running configuration to the startup configuration) on the Catalyst 3850 yet.

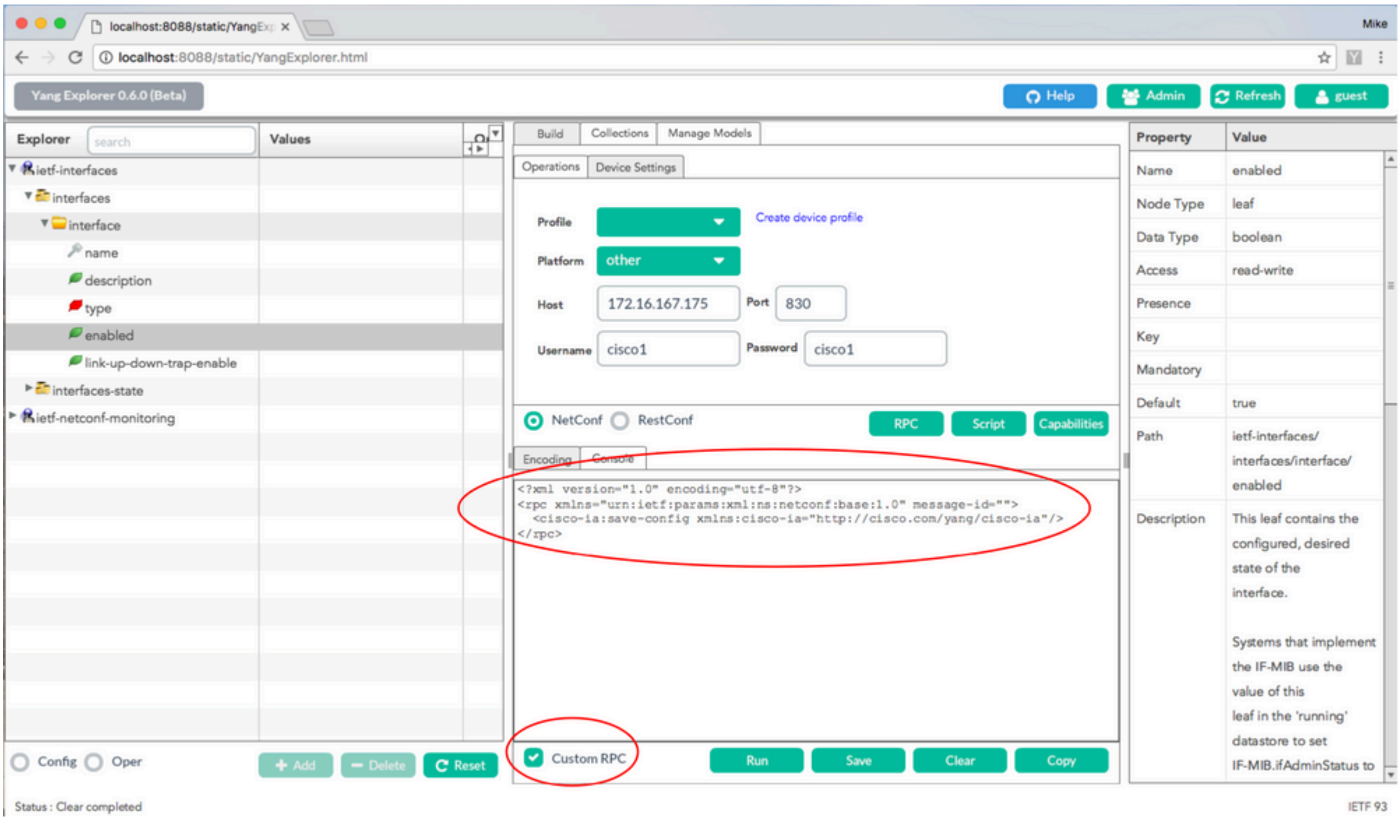
```
3850-1# show startup-config | begin 1/0/16
 interface GigabitEthernet1/0/16
 !
```

Save the Configuration on a Catalyst 3850

The running configuration can be saved to the startup configuration on the Catalyst 3850 by sending this YANG formatted NETCONF RPC message to the Catalyst 3850 via NETCONF.

```
<?xml version="1.0" encoding="utf-8"?>
 <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
   <cisco-ia:save-config xmlns:cisco-ia="cisco/yang/cisco-ia"
 </rpc>
```

This is done when you cut and paste this into the Yang Explorer application as a Custom RPC.



Profile:

Platform: other

Host: 172.16.167.175 Port: 830

Username: cisco1 Password: cisco1

● NetConf ○ RestConf

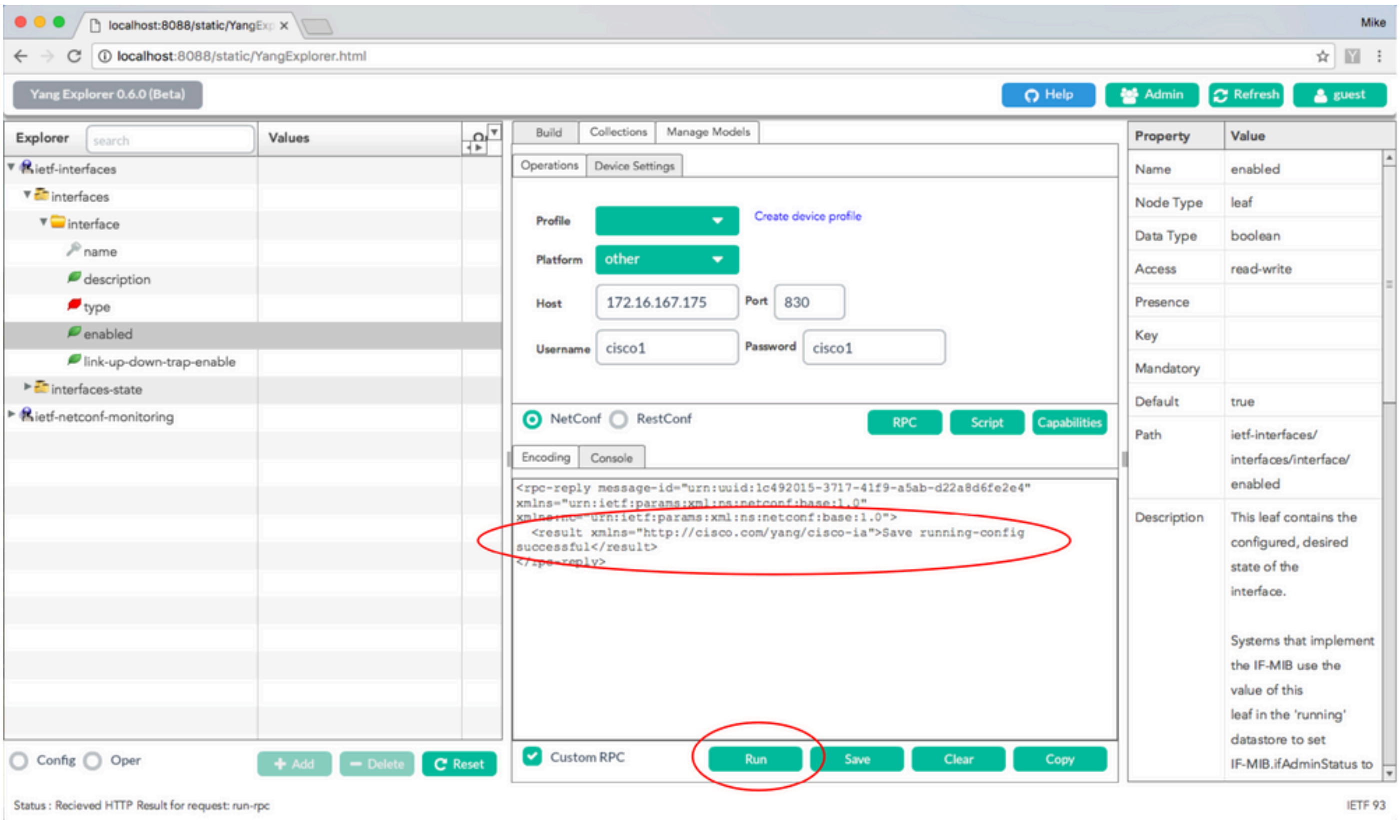
```
<?xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
  <cisco-ia:save-config xmlns:cisco-ia="http://cisco.com/yang/cisco-ia"/>
</rpc>
```

☑ Custom RPC

Property	Value
Name	enabled
Node Type	leaf
Data Type	boolean
Access	read-write
Presence	
Key	
Mandatory	
Default	true
Path	ietf-interfaces/interfaces/interface/enabled
Description	This leaf contains the configured, desired state of the interface. Systems that implement the IF-MIB use the value of this leaf in the 'running' datastore to set IF-MIB.ifAdminStatus to

Status : Clear completed

Run is selected in order to send the custom RPC message to the Catalyst 3850 via NETCONF. The Catalyst 3850 replies back with a successful message.



Profile:

Platform: other

Host: 172.16.167.175 Port: 830

Username: cisco1 Password: cisco1

● NetConf ○ RestConf

```
<rpc-reply message-id="urn:uuid:1c492015-3717-41f9-a5ab-d22a8d6fe2e4"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <result xmlns="http://cisco.com/yang/cisco-ia">Save running-config
successful</result>
</rpc-reply>
```

☑ Custom RPC

Property	Value
Name	enabled
Node Type	leaf
Data Type	boolean
Access	read-write
Presence	
Key	
Mandatory	
Default	true
Path	ietf-interfaces/interfaces/interface/enabled
Description	This leaf contains the configured, desired state of the interface. Systems that implement the IF-MIB use the value of this leaf in the 'running' datastore to set IF-MIB.ifAdminStatus to

Status : Recieved HTTP Result for request: run-rpc

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

The startup configuration now matches the running configuration:

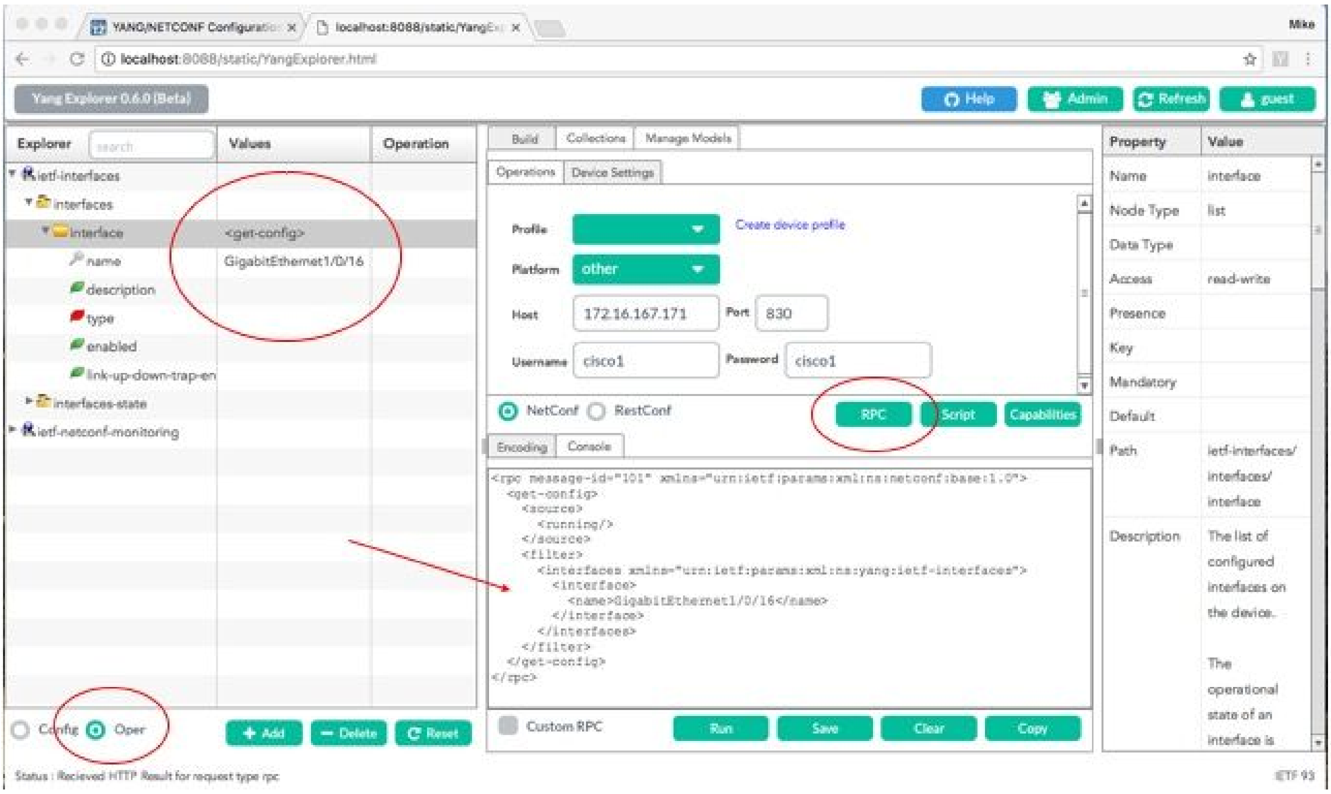
```
3850-1# show running-config interface gigabitEthernet 1/0/16
 Building configuration...
Current configuration : 49 bytes
 !
 interface GigabitEthernet1/0/16
 shutdown
 end

3850-1# show startup-config | begin 1/0/16
 interface GigabitEthernet1/0/16
 shutdown
 !
```

Configure the Catalyst 3850 from the CLI

As mentioned previously, the regular Catalyst 3850 CLI can still be used to configure the switch and collect show command data in addition to using NETCONF/YANG to do the same. When the Catalyst 3850 CLI is used instead of NETCONF/YANG to configure the switch, the new running-config is synchronized with the Data Model Interface (DMI) on the Catalyst 3850 via the syncfd software process.

```
3850-1# show running-config interface gigabitEthernet 1/0/16
Building configuration...
Current configuration : 49 bytes
!
interface GigabitEthernet1/0/16
 shutdown
end

3850-1# config t
Enter configuration commands, one per line.  End with CNTL/Z.
3850-1(config)# interface gigabitEthernet 1/0/16
3850-1(config-if)#no shutdown
3850-1(config-if)# exit
3850-1(config)# exit
3850-1#
*Jan 24 16:39:09.968: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/16, changed state to down
*Jan 24 16:39:13.479: %SYS-5-CONFIG_I: Configured from console by console
*Jan 24 16:39:15.208: %DMI-5-SYNC_START:Switch 1 R0/0: syncfd:  External change to running configuratio
*Jan 24 16:39:43.290: %DMI-5-SYNC_COMPLETE:Switch 1 R0/0: syncfd:  The running configuration has been s
3850-1#
```

The next time the Yang Explorer application requests a copy of the interface configuration after the CLI change, the change is reflected properly in the YANG output.



interface <get-config>

name GigabitEthernet1/0/16

Platform other

Host 172.16.167.171 Port 830

Username cisco1 Password cisco1

◉ NetConf ○ RestConf

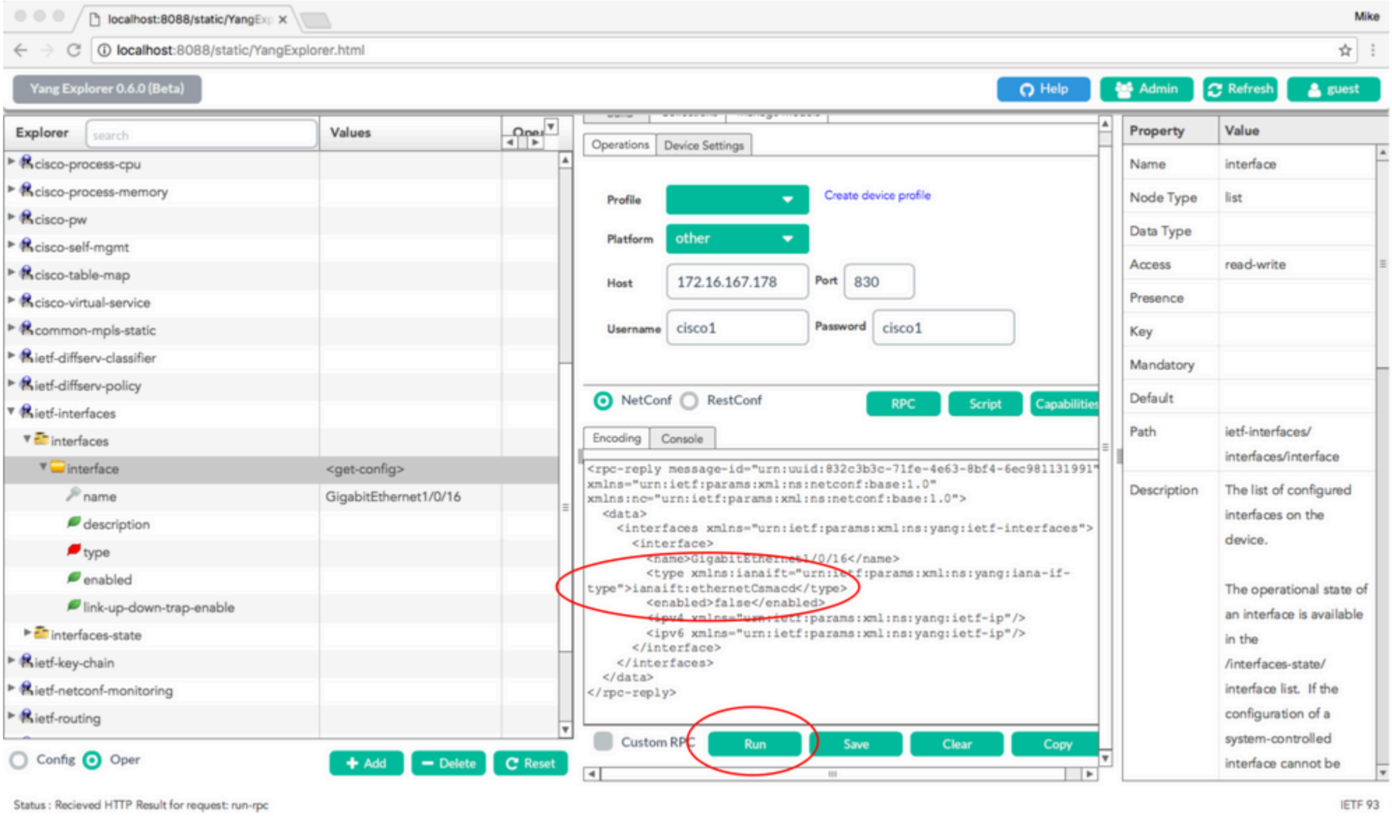
○ Config ◉ Oper

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name>GigabitEthernet1/0/16</name>
        </interface>
      </interfaces>
    </filter>
  </get-config>
</rpc>
```

Property	Value
Name	interface
Node Type	list
Data Type	
Access	read-write
Presence	
Key	
Mandatory	
Default	
Path	ietf-interfaces/ interfaces/ interface
Description	The list of configured interfaces on the device. The operational state of an interface is

Status : Recieved HTTP Result for request type rpc

Run is selected in order to send the RPC get-config message for GigabitEthernet1/0/16 to the Catalyst 3850 via NETCONF. The Catalyst 3850 replies back with the GigabitEthernet1/0/16 interface configuration which shows enabled = true.



interface <get-config>

name GigabitEthernet1/0/16

Platform other

Host 172.16.167.178 Port 830

Username cisco1 Password cisco1

◉ NetConf ○ RestConf

○ Config ◉ Oper

```
<rpc-reply message-id="urn:uuid:832c3b3c-71fe-4e63-8bf4-6ec981131991"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
      <interface>
        <name>GigabitEthernet1/0/16</name>
        <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-
type">ianaift:ethernetCsmacd</type>
        <enabled>false</enabled>
        <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"/>
        <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"/>
      </interface>
    </interfaces>
  </data>
</rpc-reply>
```

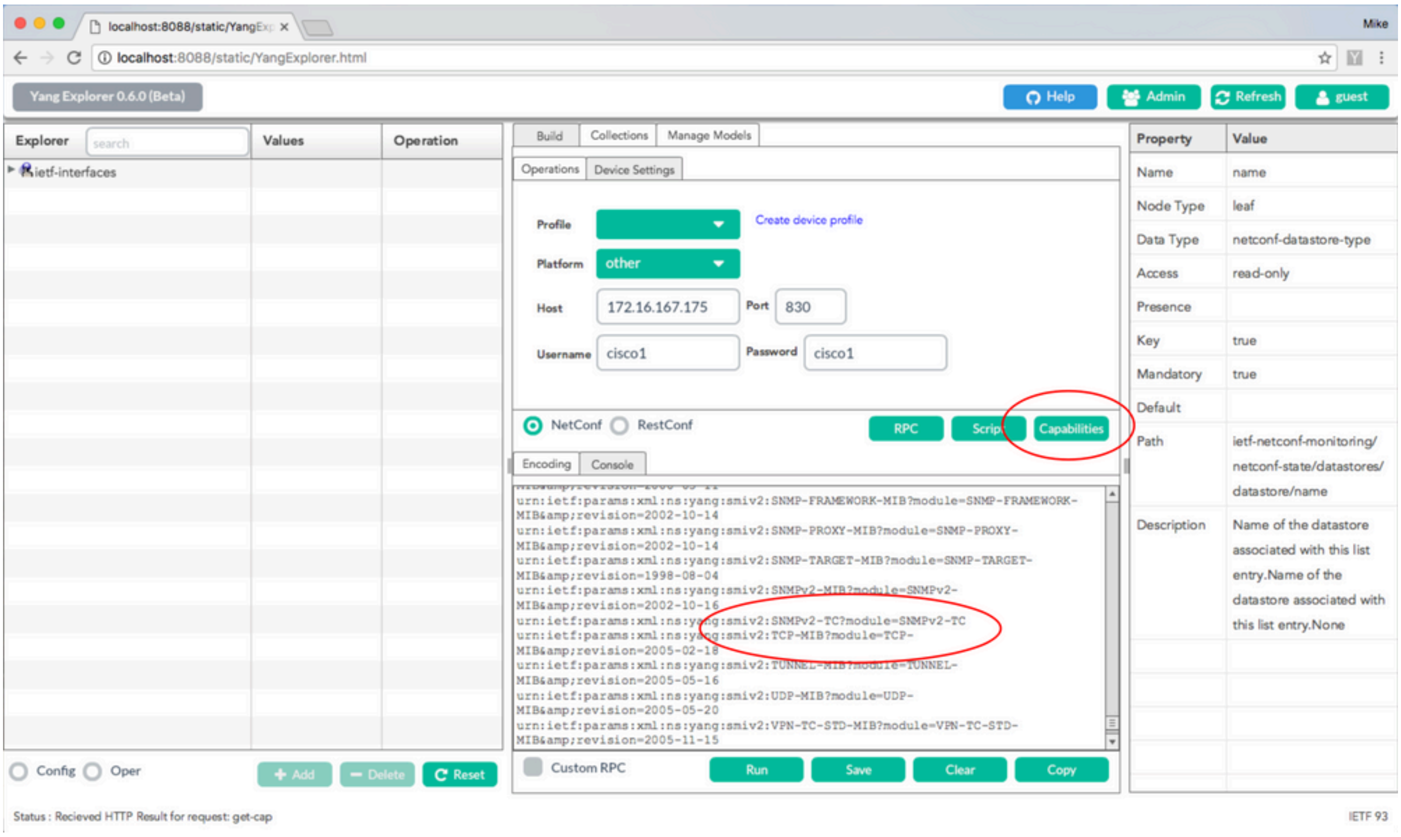
Property	Value
Name	interface
Node Type	list
Data Type	
Access	read-write
Presence	
Key	
Mandatory	
Default	
Path	ietf-interfaces/ interfaces/interface
Description	The list of configured interfaces on the device. The operational state of an interface is available in the /interfaces-state/ interface list. If the configuration of a system-controlled interface cannot be

Status : Recieved HTTP Result for request: run-rpc

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

The SNMP MIB data that can be returned with NETCONF GET operations is not user configurable. All supported SNMP MIBs that are converted into structured data defined by YANG data models are part of the Cisco XE software on the Catalyst 3850. To discover what MIB data is available in GET requests, there are three options stated. All supported MIBs can include smiv2 in the capability response.

Option 1. The **Capabilities** button can be selected in the Yang Explorer application GUI. The Catalyst 3850 replies back with its capability list that contains smiv2 MIB entries.



Profile	
Platform	other
Host	172.16.167.175
Port	830
Username	cisco1
Password	cisco1

```
urn:ietf:params:xml:ns:yang:smiv2:SNMP-FRAMEWORK-MIB?module=SNMP-FRAMEWORK-
MIB&amp;revision=2002-10-14
urn:ietf:params:xml:ns:yang:smiv2:SNMP-PROXY-MIB?module=SNMP-PROXY-
MIB&amp;revision=2002-10-14
urn:ietf:params:xml:ns:yang:smiv2:SNMP-TARGET-MIB?module=SNMP-TARGET-
MIB&amp;revision=1998-08-04
urn:ietf:params:xml:ns:yang:smiv2:SNMPv2-MIB?module=SNMPv2-
MIB&amp;revision=2002-10-16
urn:ietf:params:xml:ns:yang:smiv2:SNMPv2-TC?module=SNMPv2-TC
urn:ietf:params:xml:ns:yang:smiv2:TCP-MIB?module=TCP-
MIB&amp;revision=2005-02-18
urn:ietf:params:xml:ns:yang:smiv2:TUNNEL-MIB?module=TUNNEL-
MIB&amp;revision=2005-05-16
urn:ietf:params:xml:ns:yang:smiv2:UDP-MIB?module=UDP-
MIB&amp;revision=2005-05-20
urn:ietf:params:xml:ns:yang:smiv2:VPN-TC-STD-MIB?module=VPN-TC-STD-
MIB&amp;revision=2005-11-15
```

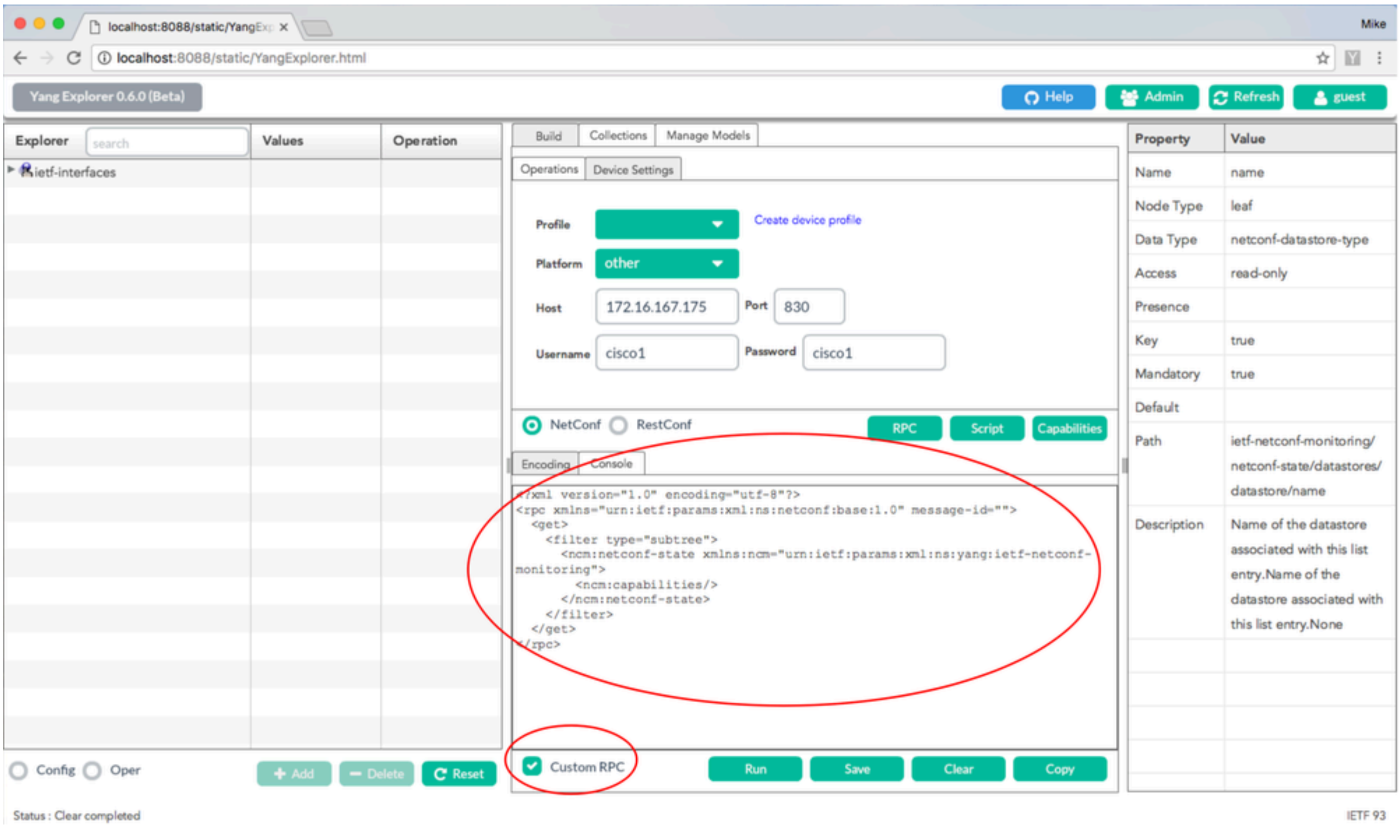
Property	Value
Name	name
Node Type	leaf
Data Type	netconf-datastore-type
Access	read-only
Presence	
Key	true
Mandatory	true
Default	
Path	ietf-netconf-monitoring/netconf-state/datastores/datastore/name
Description	Name of the datastore associated with this list entry.Name of the datastore associated with this list entry.None

Status : Recieved HTTP Result for request: get-cap

Option 2. This YANG formatted NETCONF RPC message can be sent to the Catalyst 3850 via NETCONF in order to retrieve the capabilities list which includes available smiv2 MIB models.

```
<?xml version="1.0" encoding="utf-8"?>
 <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
   <get>
     <filter type="subtree">
       <ncm:netconf-state xmlns:ncm="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
         <ncm:capabilities/>
       </ncm:netconf-state>
     </filter>
   </get>
 </rpc>
```

This is done when you cut and paste into the Yang Explorer application as a Custom RPC.



Profile

Platform: other

Host: 172.16.167.175 Port: 830

Username: cisco1 Password: cisco1

● NetConf ○ RestConf

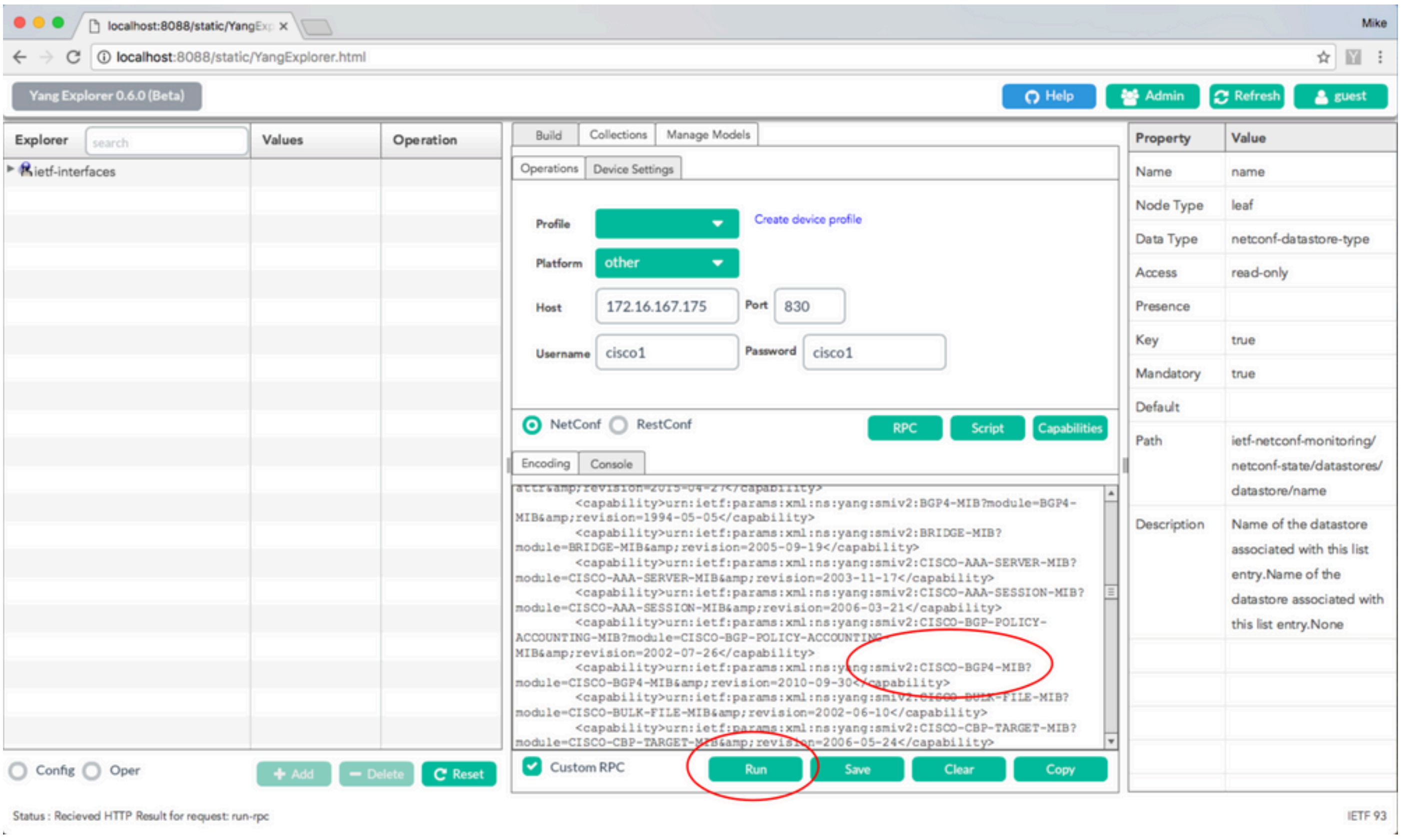
```
<?xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
  <get>
    <filter type="subtree">
      <ncm:netconf-state xmlns:ncm="urn:ietf:params:xml:ns:yang:ietf-netconf-
monitoring">
        <ncm:capabilities/>
      </ncm:netconf-state>
    </filter>
  </get>
</rpc>
```

☑ Custom RPC

Property	Value
Name	name
Node Type	leaf
Data Type	netconf-datastore-type
Access	read-only
Presence	
Key	true
Mandatory	true
Default	
Path	ietf-netconf-monitoring/netconf-state/datastores/datastore/name
Description	Name of the datastore associated with this list entry.Name of the datastore associated with this list entry.None

Status : Clear completed

Run is selected in order to send the custom RPC message to the Catalyst 3850 via NETCONF. The Catalyst 3850 replies back with a capability list that includes the smiv2 MIBs supported.



Profile

Platform: other

Host: 172.16.167.175 Port: 830

Username: cisco1 Password: cisco1

● NetConf ○ RestConf

```
attr&amp;revision=2015-04-27</capability>
        <capability>urn:ietf:params:xml:ns:yang:smiv2:BGP4-MIB?module=BGP4-
MIB&amp;revision=1994-05-05</capability>
        <capability>urn:ietf:params:xml:ns:yang:smiv2:BRIDGE-MIB?
module=BRIDGE-MIB&amp;revision=2005-09-19</capability>
        <capability>urn:ietf:params:xml:ns:yang:smiv2:CISCO-AAA-SERVER-MIB?
module=CISCO-AAA-SERVER-MIB&amp;revision=2003-11-17</capability>
        <capability>urn:ietf:params:xml:ns:yang:smiv2:CISCO-AAA-SESSION-MIB?
module=CISCO-AAA-SESSION-MIB&amp;revision=2006-03-21</capability>
        <capability>urn:ietf:params:xml:ns:yang:smiv2:CISCO-BGP-POLICY-
ACCOUNTING-MIB?module=CISCO-BGP-POLICY-ACCOUNTING-
MIB&amp;revision=2002-07-26</capability>
        <capability>urn:ietf:params:xml:ns:yang:smiv2:CISCO-BGP4-MIB?
module=CISCO-BGP4-MIB&amp;revision=2010-09-30</capability>
        <capability>urn:ietf:params:xml:ns:yang:smiv2:CISCO-BULK-FILE-MIB?
module=CISCO-BULK-FILE-MIB&amp;revision=2002-06-10</capability>
        <capability>urn:ietf:params:xml:ns:yang:smiv2:CISCO-CBP-TARGET-MIB?
module=CISCO-CBP-TARGET-MIB&amp;revision=2006-05-24</capability>
```

☑ Custom RPC

Property	Value
Name	name
Node Type	leaf
Data Type	netconf-datastore-type
Access	read-only
Presence	
Key	true
Mandatory	true
Default	
Path	ietf-netconf-monitoring/netconf-state/datastores/datastore/name
Description	Name of the datastore associated with this list entry.Name of the datastore associated with this list entry.None

Status : Recieved HTTP Result for request: run-rpc

Option 3. A list of available MIB models can be viewed in the NETCONF capabilities and Hello message returned by the Catalyst 3850 in response to an SSH connection from the Centralized Management Platform (Laptop).

```
USER1-M-902T:~ USER1$ ssh -s cisco1@172.16.167.175 -p 830 netconf
cisco1@172.16.167.175's password: cisco1

<?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:xpath:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.1</capability>
<capability>urn:ietf:params:netconf:capability:rollback-on-error:1.0</capability
--snip--
<capability>urn:ietf:params:xml:ns:yang:smiv2:CISCO-CONFIG-MAN-MIB?module=CISCO-CONFIG-MAN-MIB&amp;revi
<capability>urn:ietf:params:xml:ns:yang:smiv2:CISCO-CONTEXT-MAPPING-MIB?module=CISCO-CONTEXT-MAPPING-MI
<capability>urn:ietf:params:xml:ns:yang:smiv2:CISCO-DATA-COLLECTION-MIB?module=CISCO-DATA-COLLECTION-MI
--snip--
</capabilities>
<session-id>2870</session-id></ hello >]]>]]>

Use < ^C > to exit
```

Load Additional YANG Data Models

This link contains additional YANG data model files. These files allow additional operations to be executed via NETCONF/YANG that relates to other Catalyst 3850 features such as to configure IPv4 unicast routing, QoS, and so on.

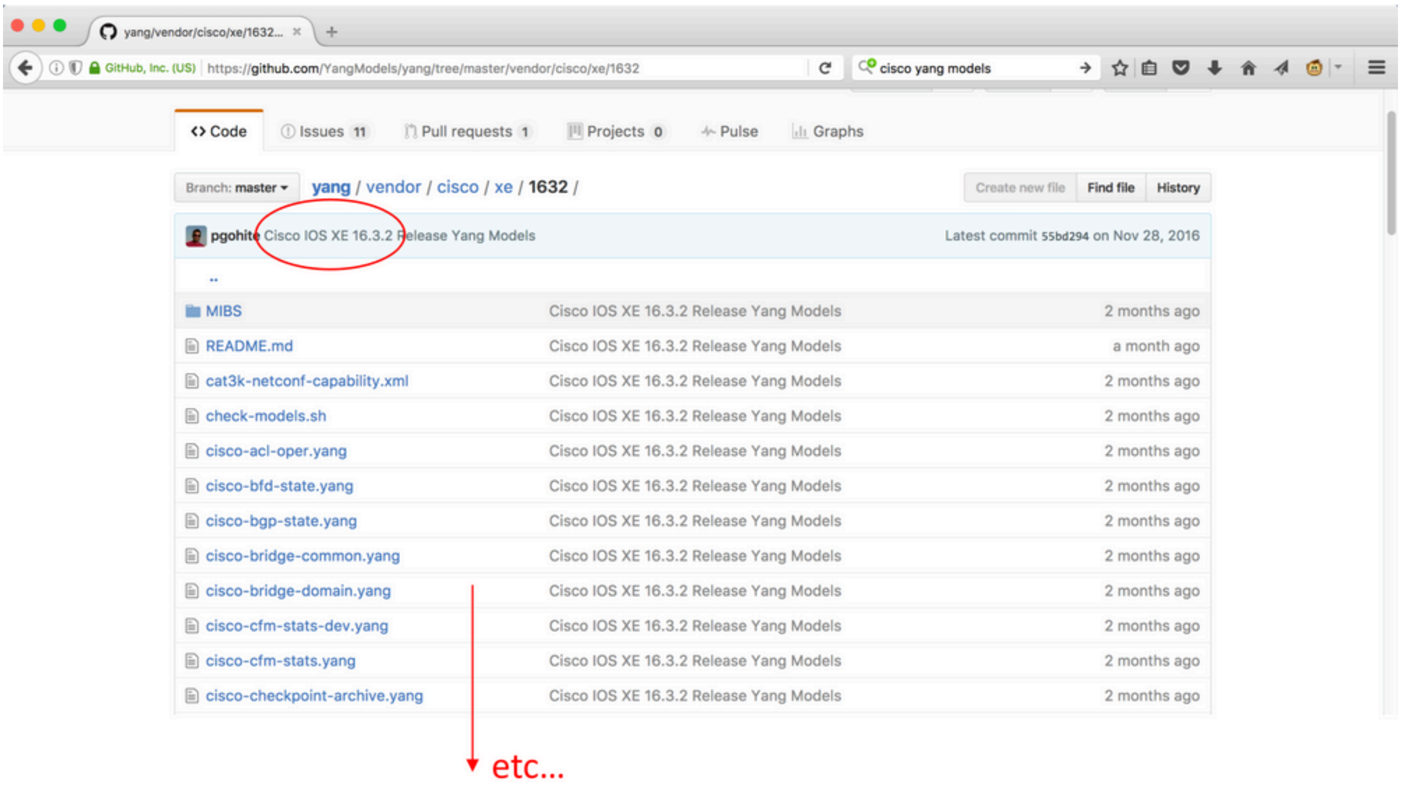
GitHub Yang Models

The standard (common, Internet Engineering Task Force (IETF)) models that apply to all vendors can be found by choosing **standard, ietf, rfc**. This provides the standards based YANG data models taken from RFC publications by the IETF standards body.

GitHub Yang Models Tree Master Standard

The Cisco native (device, vendor specific) models can be found by selecting **vendor, cisco, xe, 1632**. This provides the proprietary YANG data models for Cisco IOS XE software version 16.3.2 for the Catalyst 3850.

GitHub Yang Models Yang Tree Master Vendor



https://github.com/YangModels/yang/tree/master/vendor/cisco/xe/1632

yang / vendor / cisco / xe / 1632 /

pgohite Cisco IOS XE 16.3.2 Release Yang Models — Latest commit 55bd294 on Nov 28, 2016

..		
MIBS	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
README.md	Cisco IOS XE 16.3.2 Release Yang Models	a month ago
cat3k-netconf-capability.xml	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
check-models.sh	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
cisco-acl-oper.yang	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
cisco-bfd-state.yang	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
cisco-bgp-state.yang	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
cisco-bridge-common.yang	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
cisco-bridge-domain.yang	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
cisco-cfm-stats-dev.yang	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
cisco-cfm-stats.yang	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
cisco-checkpoint-archive.yang	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago

etc...

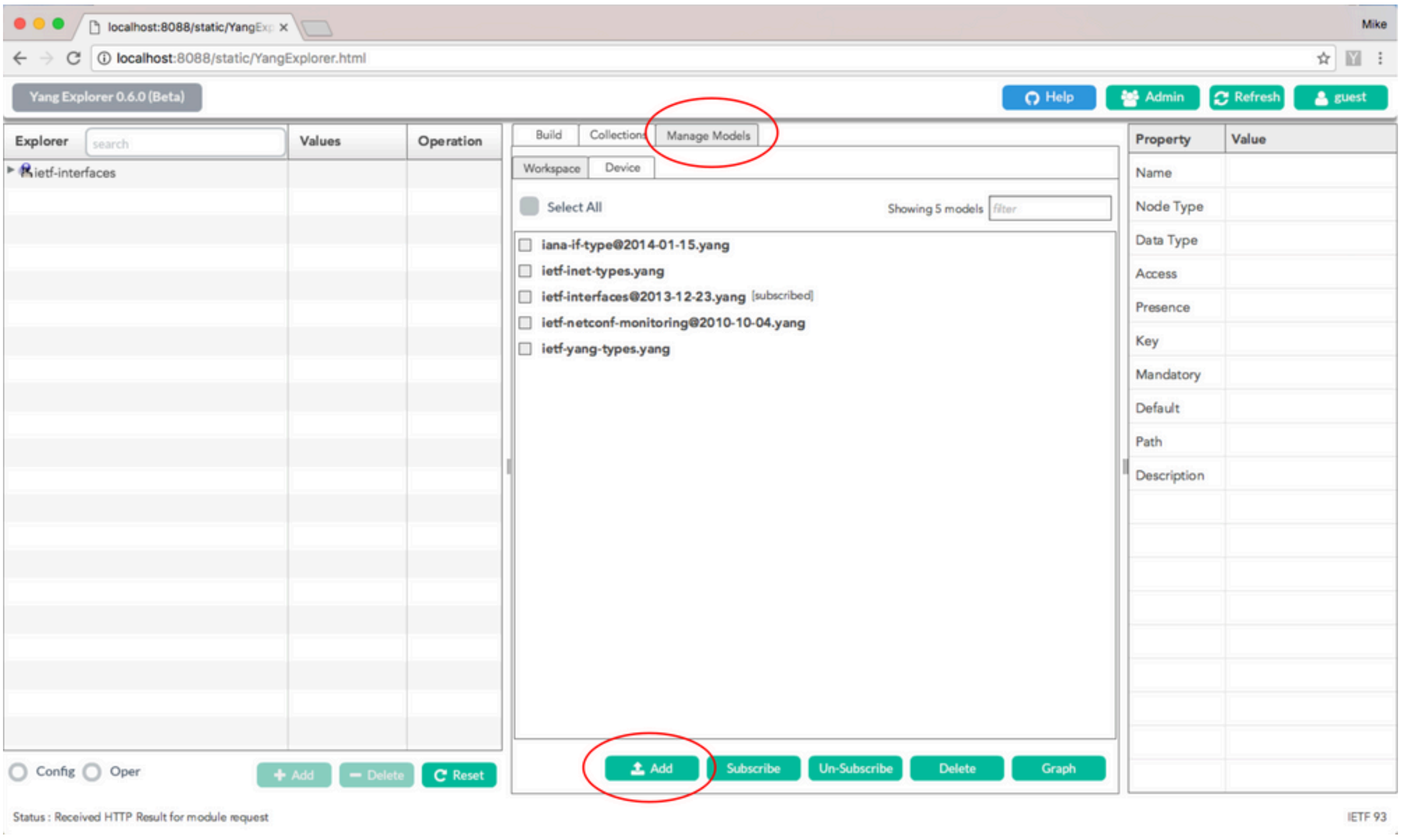
These files can be downloaded onto the Centralized Management Platform (laptop) and then, in turn loaded into the Yang Explorer application. There are two ways to do this. The first is to load in the various YANG data model files individually, the second is a bulk loading of all the files.

1. Load the Various YANG Data Model Files Individually

Tip: rawgit can be required to download the files from Github. To download files from github select the **Raw** button associated with the YANG file. If a URL is given instead of a file download option, the URL can be pasted into rawgit which can in turn provide a production URL. Paste this new production URL into a browser and it can provide the file download option.

In this example, cisco-ethernet.yang has already been downloaded from github onto the Centralized Management Platform (laptop). Here are the steps to load the file into the Yang Explorer application GUI and then **Subscribe** to it so that it is loaded into the Explorer section of the tool.

Tip: NETCONF capabilities functionality can be used to determine which data models are supported by the Catalyst 3850 software. See section 2. of Configuring the Centralized Management Platform (laptop).



□ iana-if-type@2014-01-15.yang

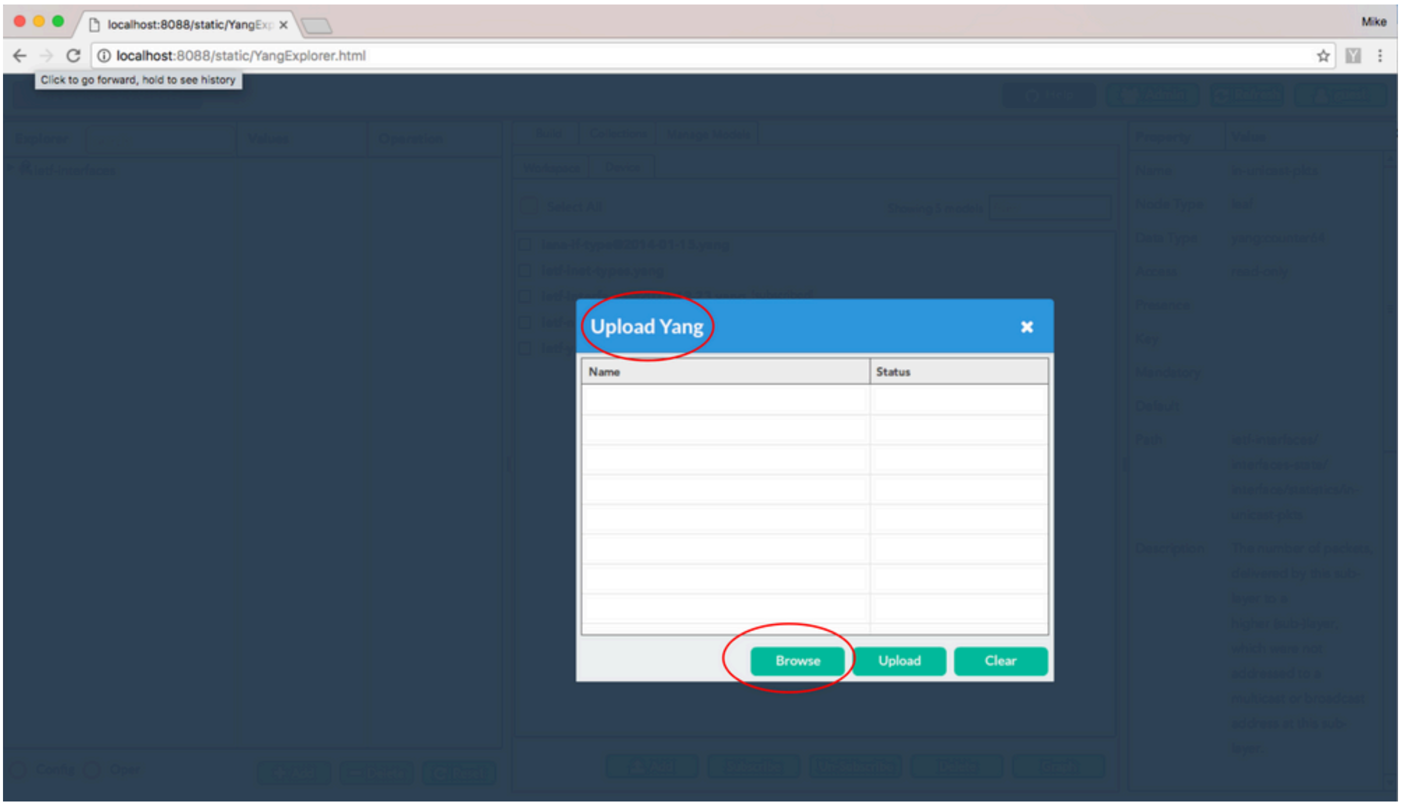
□ ietf-inet-types.yang

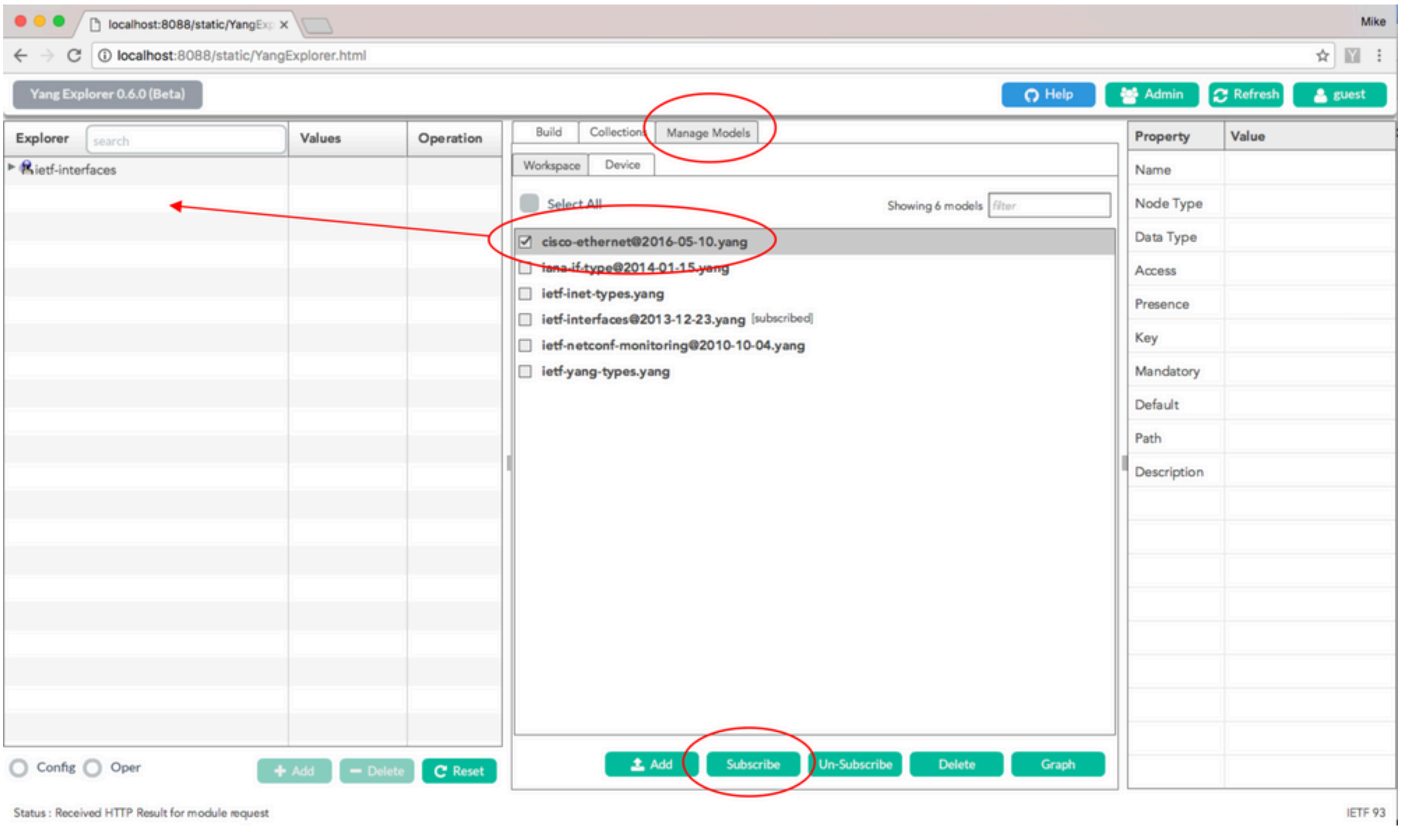
□ ietf-interfaces@2013-12-23.yang [subscribed]

□ ietf-netconf-monitoring@2010-10-04.yang

□ ietf-yang-types.yang

Status : Received HTTP Result for module request


localhost:8088/static/YangExp
Mike
localhost:8088/static/YangExplorer.html
Click to go forward, hold to see history
Upload Yang
Name
Status
Browse
Upload
Clear



□ Select All

Showing 6 models filter

☑ cisco-ethernet@2016-05-10.yang
□ iana-if-type@2014-01-15.yang
□ ietf-inet-types.yang
□ ietf-interfaces@2013-12-23.yang [subscribed]
□ ietf-netconf-monitoring@2010-10-04.yang
□ ietf-yang-types.yang

Status : Received HTTP Result for module request

2. Bulk Load of All the YANG Data Model Files at Once

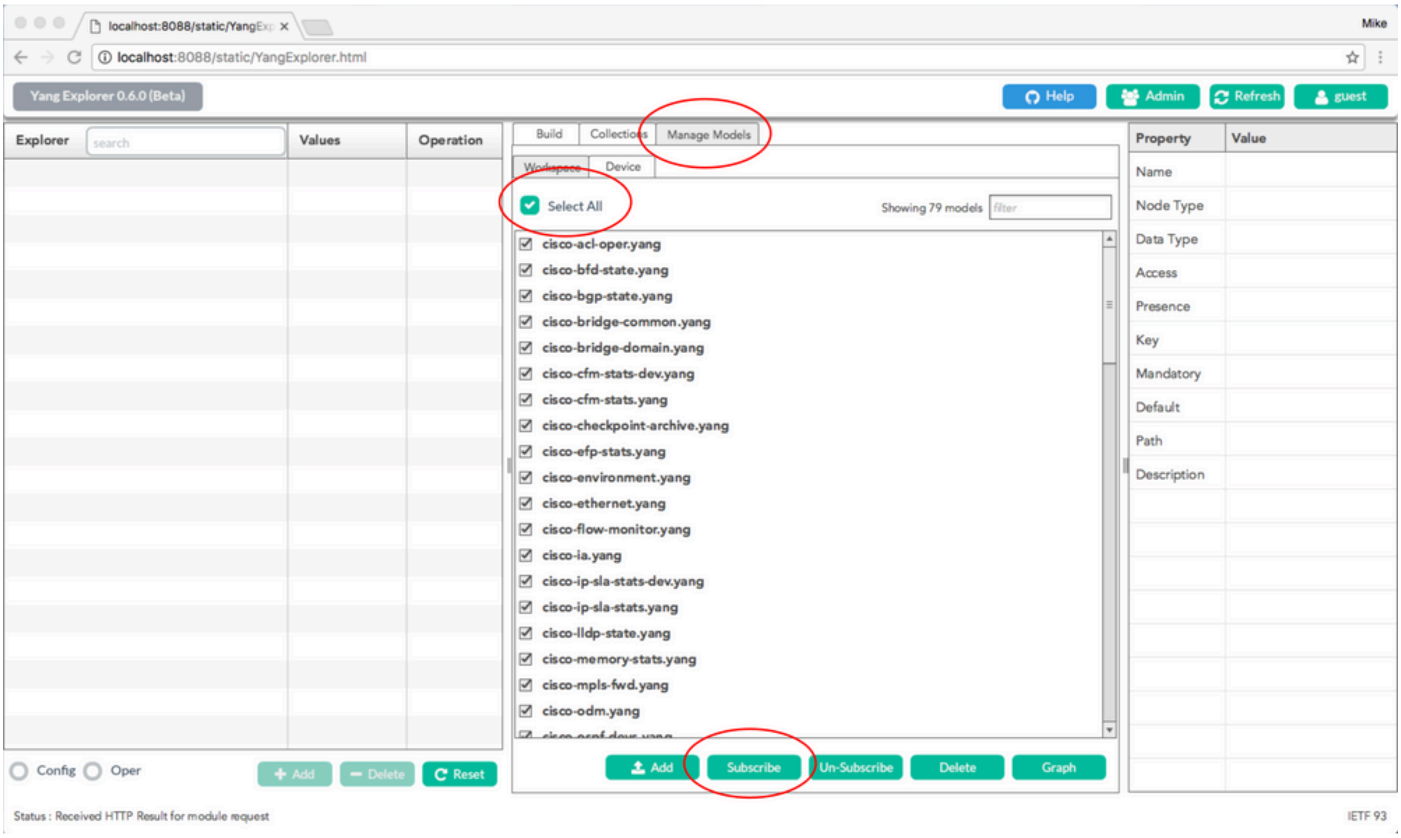
This procedure is also mentioned in section 5.2.2 here: github.

From a terminal prompt on the centralized management platform (laptop - Apple MacBook Pro running macOS Sierra 10.12.2):

```
USER1-M-902T:~ USER1$ cd yang-explorer □
USER1-M-902T:yang-explorer USER1$ cd server □
USER1-M-902T:server USER1$ python manage.py bulkupload --user guest --git https://github.com/YangModels

Git upload .. □
Cloning into '/Users/USER1/yang-explorer/server/data/session/tmpk7V406'...
□remote: Counting objects: 5610, done. □
remote: Total 5610 (delta 0), reused 0 (delta 0), pack-reused 5610 □
Receiving objects: 100% (5610/5610), 11.80 MiB | 2.34 MiB/s, done. □
Resolving deltas: 100% (3159/3159), done. □
Checking out files: 100% (3529/3529), done.
□Cleaning up /Users/USER1/yang-explorer/server/data/session/tmpk7V406 □
Compiling : user: guest, file: /Users/USER1/yang-explorer/server/data/session/tmpHTAEP3/cisco-acl-oper.
□DEBUG:root:Compiling session dependency ...
□//anaconda/bin/pyang □
DEBUG:root:Rebuilding dependencies for user guest
□--snip--
```

All of the Yang data models are now seen in the Yang Explorer application GUI. The files associated with the features of interest can be selected when you click **Subscribe** which then adds them into the Explorer section of the tool.

Tip: NETCONF capabilities functionality can be used to determine which data models are supported by the Catalyst software. See section 2. of Configuring the Centralized Management Platform (laptop).

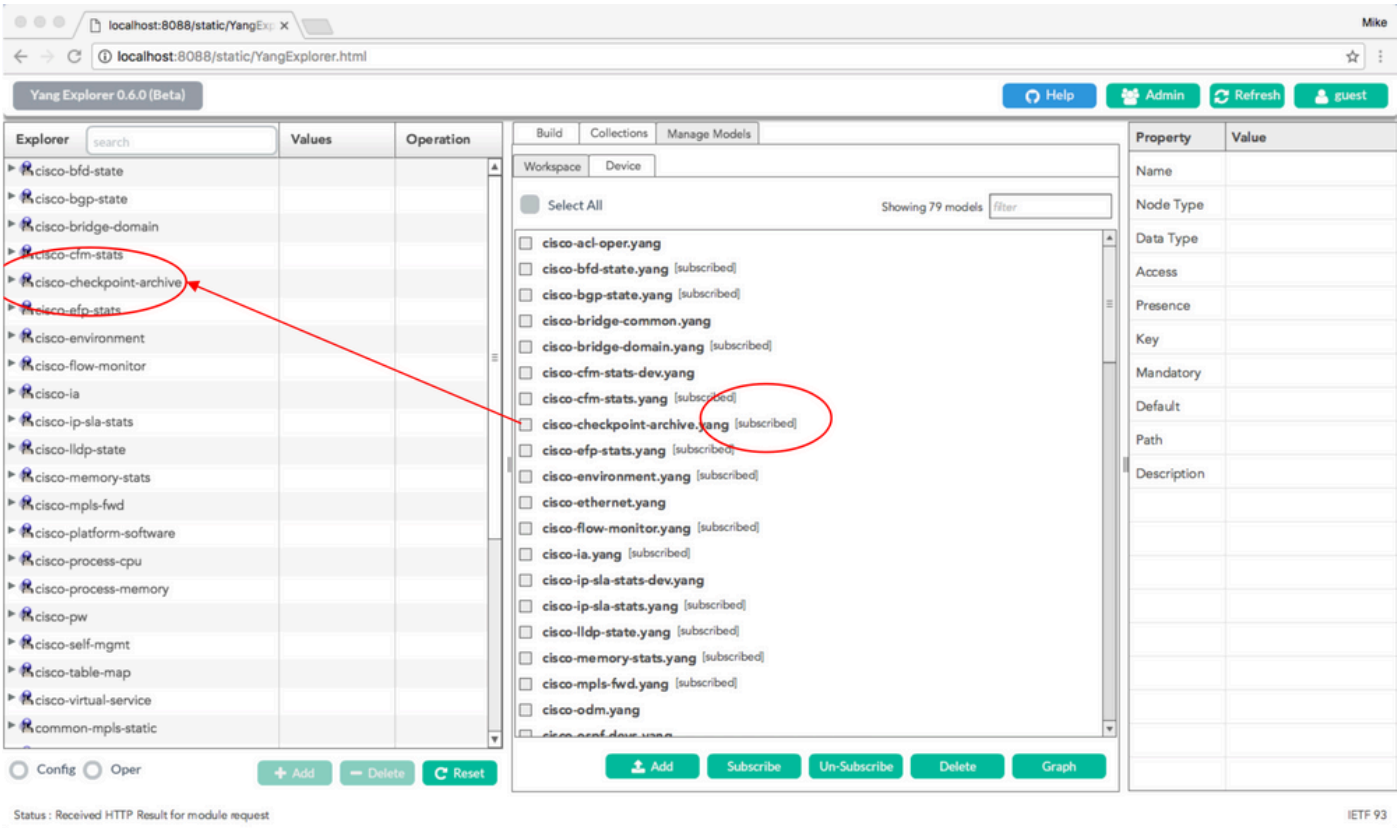


☑ Select All

Showing 79 models

☑ cisco-acl-oper.yang
☑ cisco-bfd-state.yang
☑ cisco-bgp-state.yang
☑ cisco-bridge-common.yang
☑ cisco-bridge-domain.yang
☑ cisco-cfm-stats-dev.yang
☑ cisco-cfm-stats.yang
☑ cisco-checkpoint-archive.yang
☑ cisco-efp-stats.yang
☑ cisco-environment.yang
☑ cisco-ethernet.yang
☑ cisco-flow-monitor.yang
☑ cisco-ia.yang
☑ cisco-ip-sla-stats-dev.yang
☑ cisco-ip-sla-stats.yang
☑ cisco-lldp-state.yang
☑ cisco-memory-stats.yang
☑ cisco-mpls-fwd.yang
☑ cisco-odm.yang

Status : Received HTTP Result for module request



cisco-bfd-state
cisco-bgp-state
cisco-bridge-domain
cisco-cfm-stats
cisco-checkpoint-archive
cisco-efp-stats
cisco-environment
cisco-flow-monitor
cisco-ia
cisco-ip-sla-stats
cisco-lldp-state
cisco-memory-stats
cisco-mpls-fwd
cisco-platform-software
cisco-process-cpu
cisco-process-memory
cisco-pw
cisco-self-mgmt
cisco-table-map
cisco-virtual-service
common-mpls-static

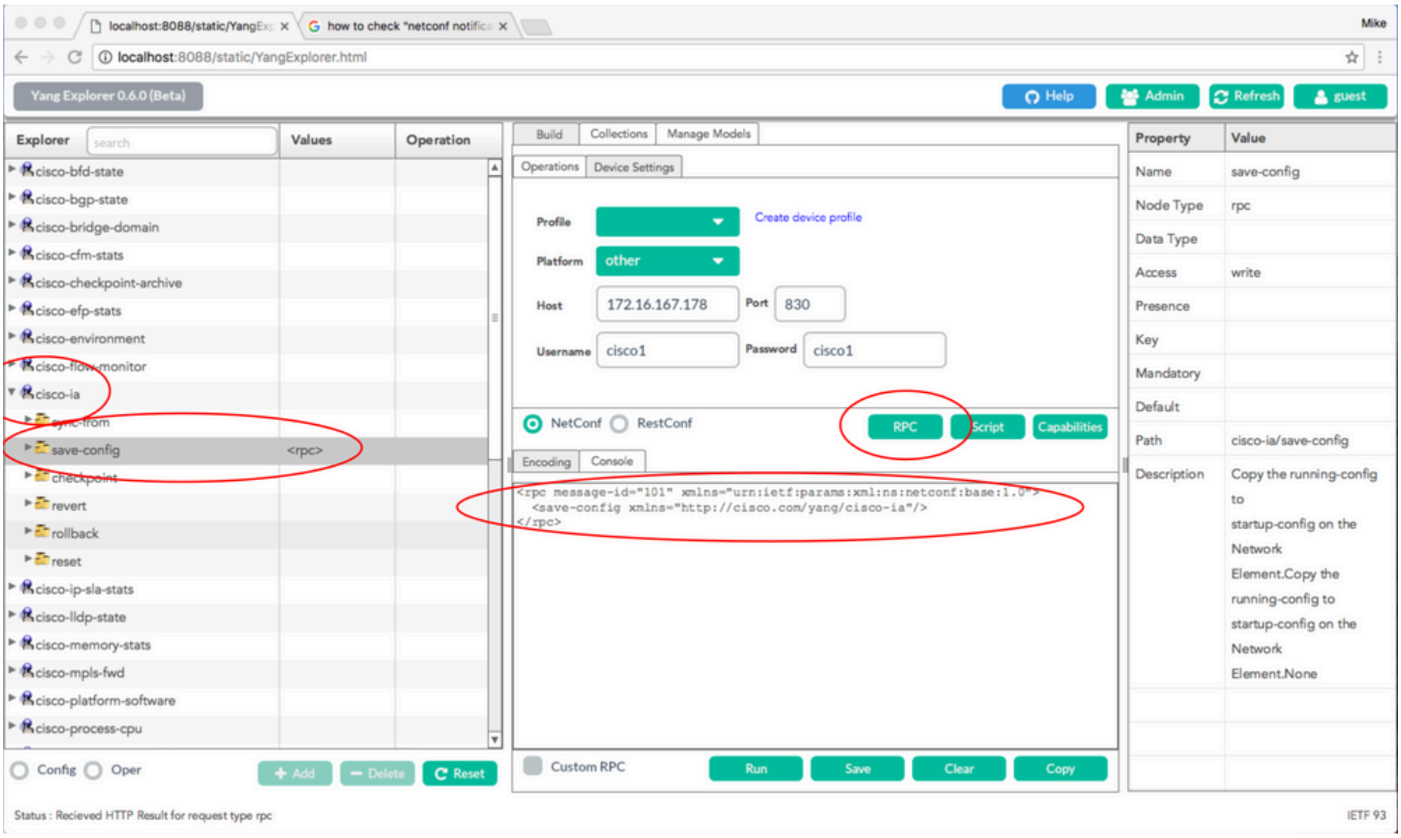
□ Select All

Showing 79 models

□ cisco-acl-oper.yang
□ cisco-bfd-state.yang [subscribed]
□ cisco-bgp-state.yang [subscribed]
□ cisco-bridge-common.yang
□ cisco-bridge-domain.yang [subscribed]
□ cisco-cfm-stats-dev.yang
□ cisco-cfm-stats.yang [subscribed]
□ cisco-checkpoint-archive.yang [subscribed]
□ cisco-efp-stats.yang [subscribed]
□ cisco-environment.yang [subscribed]
□ cisco-ethernet.yang
□ cisco-flow-monitor.yang [subscribed]
□ cisco-ia.yang [subscribed]
□ cisco-ip-sla-stats-dev.yang
□ cisco-ip-sla-stats.yang [subscribed]
□ cisco-lldp-state.yang [subscribed]
□ cisco-memory-stats.yang [subscribed]
□ cisco-mpls-fwd.yang [subscribed]
□ cisco-odm.yang

Status : Received HTTP Result for module request

Other tasks can now be completed such as to generate the NETCONF/YANG RPC required to save the configuration on the Catalyst 3850. This is done when you select the **save-conf** RPC in the Explorer section on the left hand side of the Yang Explorer application. Then, **RPC** is selected to generate the YANG formatted NETCONF RPC that can be sent to the Catalyst 3850 via NETCONF to save the configuration on the Catalyst 3850.



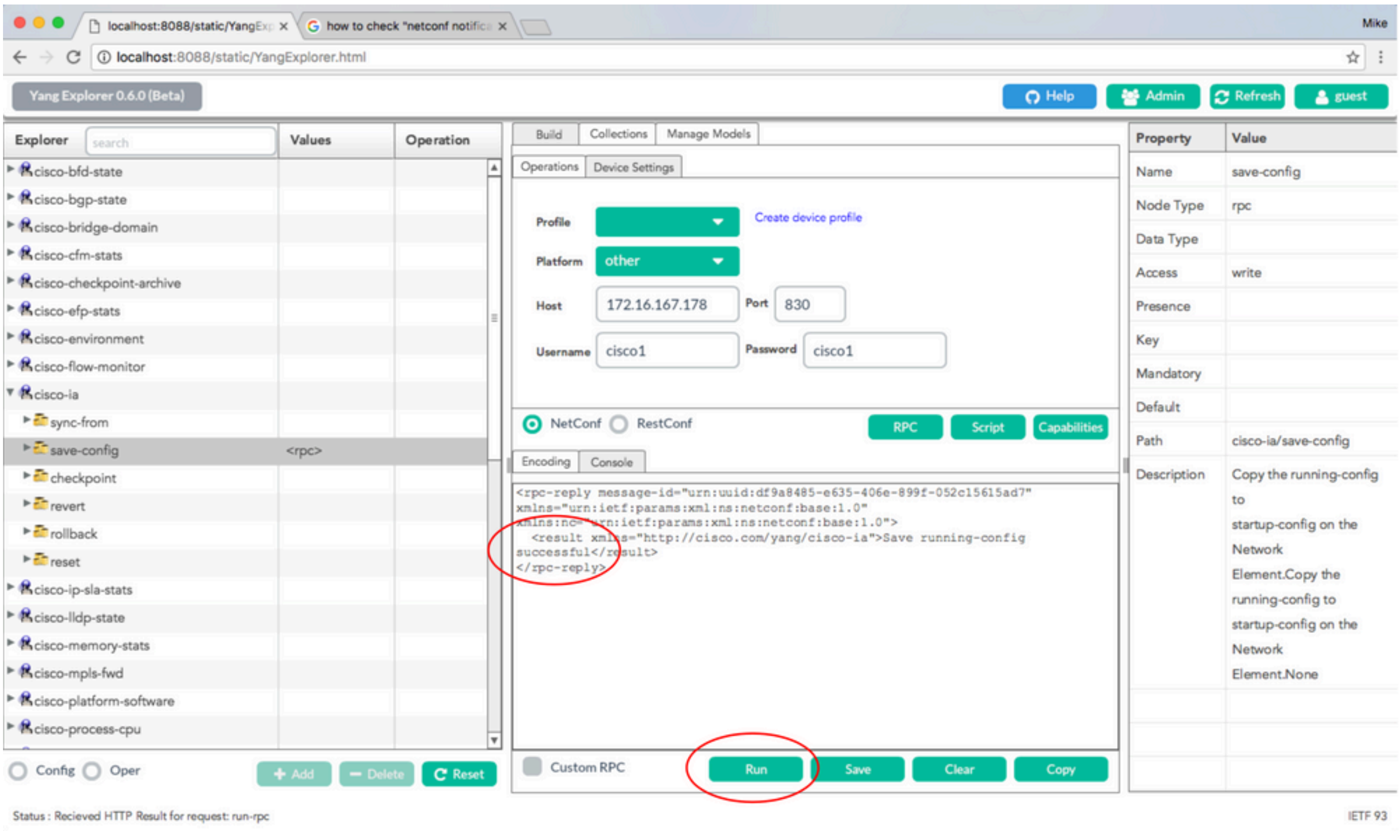
Profile:
Platform: other
Host: 172.16.167.178 Port: 830
Username: cisco1 Password: cisco1
● NetConf ○ RestConf

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-config xmlns="http://cisco.com/yang/cisco-ia"/>
</rpc>
```

Property	Value
Name	save-config
Node Type	rpc
Data Type	
Access	write
Presence	
Key	
Mandatory	
Default	
Path	cisco-ia/save-config
Description	Copy the running-config to startup-config on the Network Element.Copy the running-config to startup-config on the Network Element.None

Status : Recieved HTTP Result for request type rpc

Run is selected to send the custom RPC message to the Catalyst 3850 via NETCONF. The Catalyst 3850 replies back with a successful message.



Profile:
Platform: other
Host: 172.16.167.178 Port: 830
Username: cisco1 Password: cisco1
● NetConf ○ RestConf

```
<rpc-reply message-id="urn:uuid:df9a8485-e635-406e-899f-052c15615ad7"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <result xmlns="http://cisco.com/yang/cisco-ia">Save running-config
successful</result>
</rpc-reply>
```

Property	Value
Name	save-config
Node Type	rpc
Data Type	
Access	write
Presence	
Key	
Mandatory	
Default	
Path	cisco-ia/save-config
Description	Copy the running-config to startup-config on the Network Element.Copy the running-config to startup-config on the Network Element.None

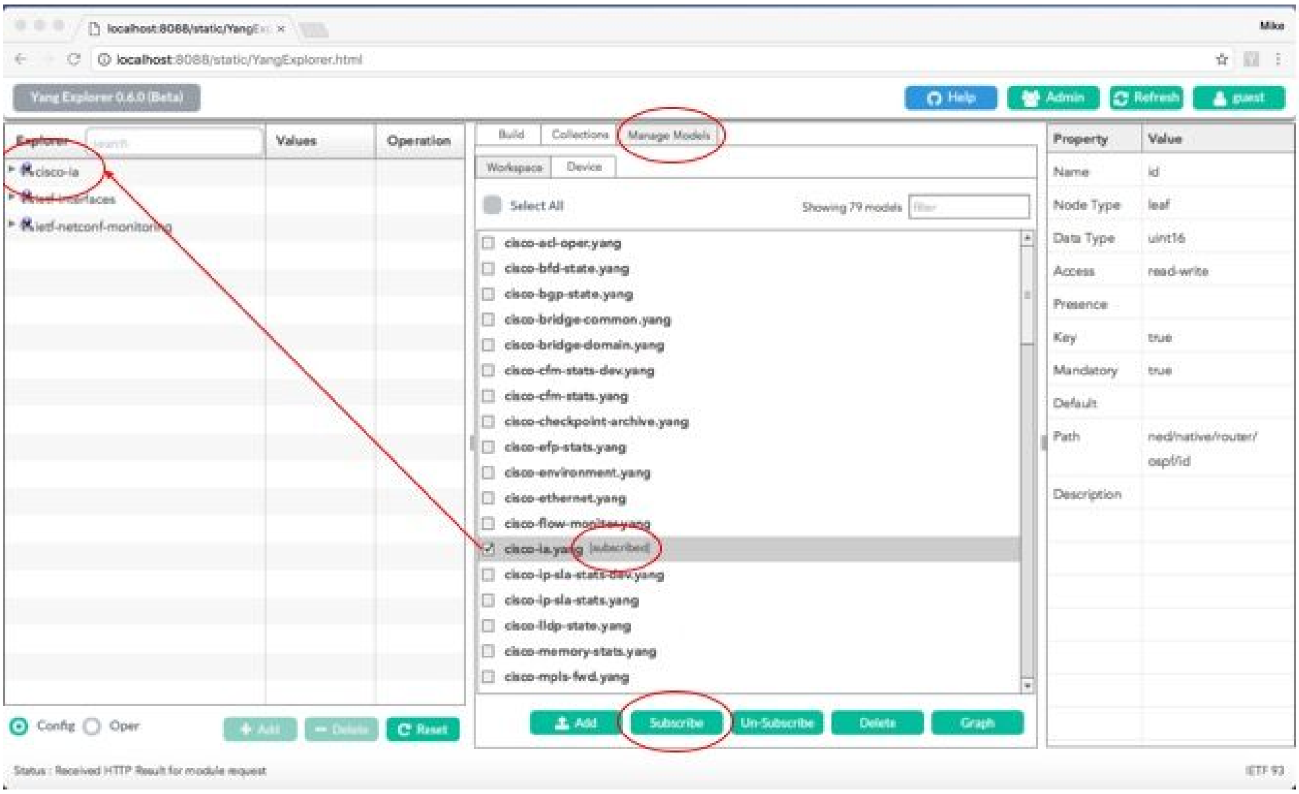
Status : Recieved HTTP Result for request: run-rpc

Notable YANG Data Models

cisco-ia.yang Data Model

Here are some RPC examples for the cisco-ia.yang data model. They are notable since they involve operations such as to save the Catalyst 3850 configuration, to synchronize the Catalyst 3850 running-config to the local Data Model Interface (DMI) data store, and reset the DMI interface on the Catalyst 3850.

The first step is to **Subscribe** to the cisco-ia.yang data model so that it appears in the Explorer section on the left of the YANG Explorer application GUI.

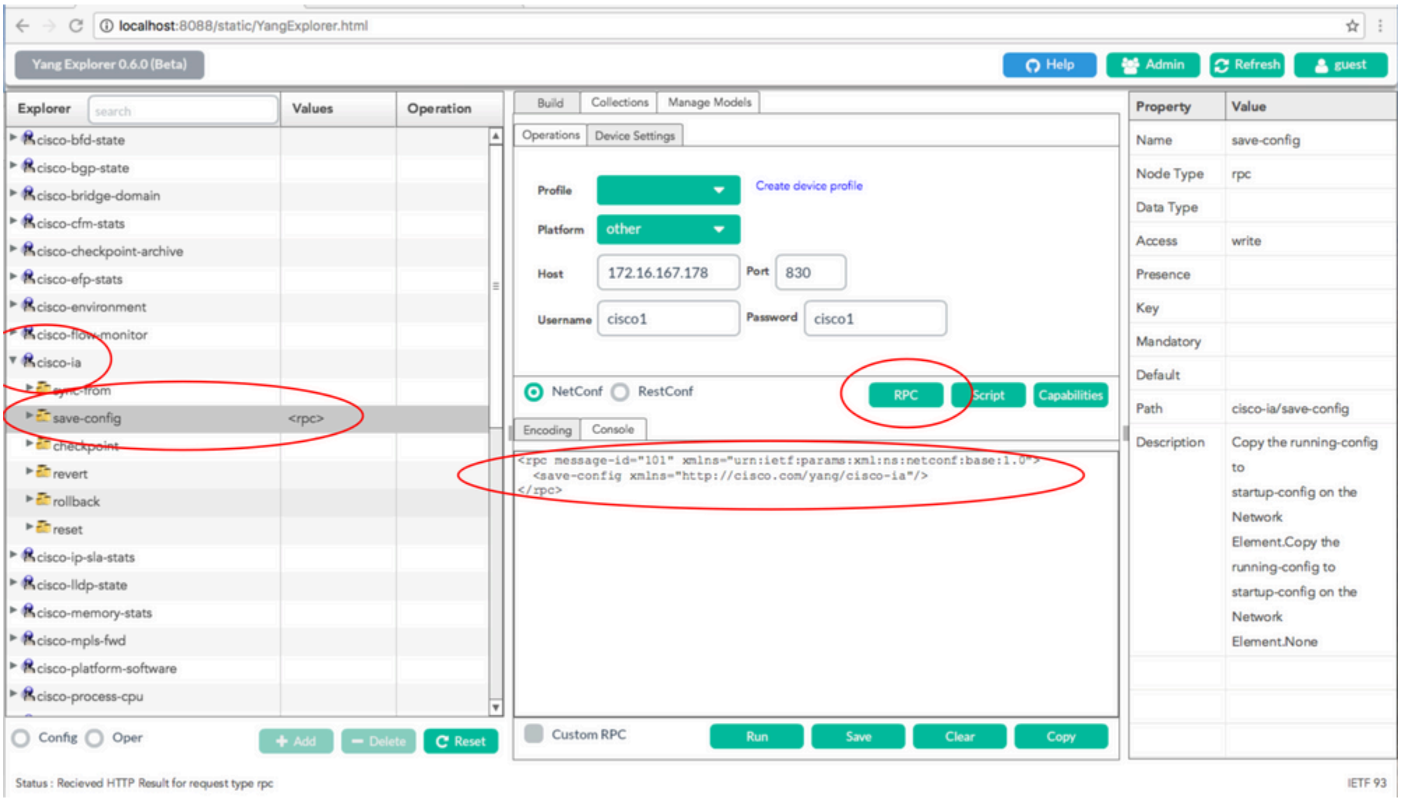


☐ cisco-acl-oper.yang
☐ cisco-bfd-state.yang
☐ cisco-bgp-state.yang
☐ cisco-bridge-common.yang
☐ cisco-bridge-domain.yang
☐ cisco-cfm-stats-dev.yang
☐ cisco-cfm-stats.yang
☐ cisco-checkpoint-archive.yang
☐ cisco-efp-stats.yang
☐ cisco-environment.yang
☐ cisco-ethernet.yang
☐ cisco-flow-monitor.yang
☑ cisco-ia.yang (subscribed)
☐ cisco-ip-sla-stats-dev.yang
☐ cisco-ip-sla-stats.yang
☐ cisco-lldp-state.yang
☐ cisco-memory-stats.yang
☐ cisco-mpls-fwd.yang

Property	Value
Name	id
Node Type	leaf
Data Type	uint16
Access	read-write
Presence	
Key	true
Mandatory	true
Default	
Path	ned/native/router/ ospf/id
Description	

Status : Received HTTP Result for module request

Once the cisco-ia data model is expanded in the Explorer section on the left of the YANG Explorer application GUI, the various operational options are seen. As an example, to use one of the available cisco-ia.yang data model options, the **save-config** operation is selected and the associated RPC is generated when you select the **RPC button**.



Profile:
Platform: other
Host: 172.16.167.178 Port: 830
Username: cisco1 Password: cisco1

● NetConf ○ RestConf

RPC Script Capabilities

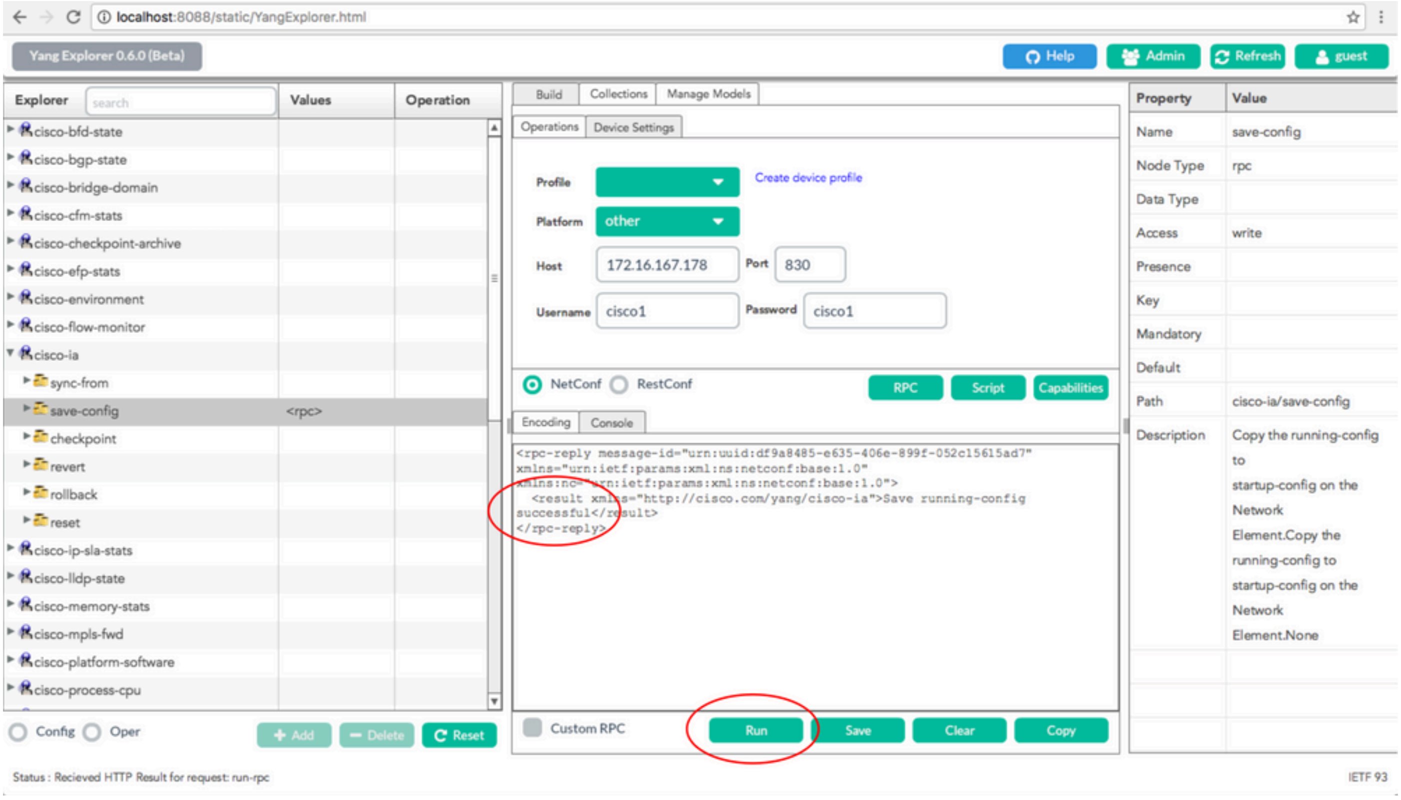
Encoding Console

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-config xmlns="http://cisco.com/yang/cisco-ia"/>
</rpc>
```

Property	Value
Name	save-config
Node Type	rpc
Data Type	
Access	write
Presence	
Key	
Mandatory	
Default	
Path	cisco-ia/save-config
Description	Copy the running-config to startup-config on the Network Element.Copy the running-config to startup-config on the Network Element.None

Status : Recieved HTTP Result for request type rpc

Next, **Run** is selected in order to send the RPC message to the Catalyst 3850 via NETCONF. The Catalyst 3850 replies back with a successful message to let the user know the operation was successful.



Profile:
Platform: other
Host: 172.16.167.178 Port: 830
Username: cisco1 Password: cisco1

● NetConf ○ RestConf

```
<rpc-reply message-id="urn:uuid:df9a8485-e635-406e-899f-052c15615ad7"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <result xmlns="http://cisco.com/yang/cisco-ia">Save running-config
successful</result>
</rpc-reply>
```

Property	Value
Name	save-config
Node Type	rpc
Data Type	
Access	write
Presence	
Key	
Mandatory	
Default	
Path	cisco-ia/save-config
Description	Copy the running-config to startup-config on the Network Element.Copy the running-config to startup-config on the Network Element.None

Status : Recieved HTTP Result for request: run-rpc

All of the various cisco-ia.yang data model operations are described here:

sync-from - This RPC causes the NETCONF interface on the Catalyst 3850 to synchronize the NETCONF datastore representation of the device running configuration with the running configuration on the device. Both of these exist on the Catalyst 3850 itself.

```
<?xml version="1.0" encoding="utf-8"?>
 <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
   <cisco-ia:sync-from xmlns:cisco-ia
 </rpc>
```

The default behavior of this RPC is to perform a sync without-defaults which causes the output of a **show running-config** command sent to the device to be synced with the NETCONF datastore. If sync-defaults is present, the NETCONF interface also reads the default configuration information provided by feature code. In most cases this option is not used. Typically this would only be used if the NETCONF interface user wanted to use the **NETCONF replace** commands to replace complete sections of the device configuration.

```
<?xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
   <cisco-ia:sync-from xmlns:cisco-ia/>
     <cisco-ia:sync-defaults/>
   </cisco-ia:sync-from>
 </rpc>
```

save-config - This RPC executes a write memory (copy running-config startup-config) command to save the current device running configuration to the device startup-configuration.

```
<?xml version="1.0" encoding="utf-8"?>
 <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
    <cisco-ia:save-config xmlns:cisco-ia
 </rpc>
```

checkpoint - This RPC causes the NETCONF interface to save the running configuration to non-volatile storage using the Cisco IOSd built-in configuration archive feature.

```
<?xml version="1.0" encoding="utf-8"?>
 <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
   <cisco-ia:checkpoint xmlns:cisco-ia
 </rpc>
```

rollback - This RPC causes the NETCONF interface to rollback the running-configuration of the device to a running configuration that was saved with the checkpoint RPC or any other valid running configuration saved on the device.

```
target-url              string (name of the saved checkpoint file)
 verbose?              Boolean (show detail during rollback process)
 nolock?                Boolean (lock configuration)
 revert-on-error?   Empty (if error occurs during rollback, leave running unchanged)
 revert-timer?        int16 (time in seconds before revets to the original configuration)
```

```
<?xml version="1.0" encoding="utf-8"?>
 <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
   <cisco-ia:rollback xmlns:cisco-ia=
     <cisco-ia:target-url>saved-config</cisco-ia:target-url>
     <cisco-ia:verbose>true</cisco-ia:verbose>
     <cisco-ia:nolock>true</cisco-ia:nolock>
     <cisco-ia:revert-on-error></cisco-ia:revert-on-error>
     <cisco-ia:revert-timer>10</cisco-ia:revert-timer>
   </cisco-ia:rollback>
 </rpc>
```

revert - This RPC causes the NETCONF interface to change the revert-timer from the rollback RPC. This cancels the timed rollback and triggers the rollback immediately, or reset parameters for the timed rollback.

```
now?   empty
timer?  int16
idle?    int16
```

```
<?xml version="1.0" encoding="utf-8"?>
 <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
   <cisco-ia:revert xmlns:cisco-ia
     <cisco-ia:now/>
     <cisco-ia:timer>10</cisco-ia:timer>
     <cisco-ia:idle>60</cisco-ia:idle>
   </cisco-ia:revert>
 </rpc>
```

reset - The NETCONF interface can be restarted with this RPC. If reinitialize is true, the NETCONF interface clears all the state information that exists in the writable-running datastore. If false (the default) the NETCONF configuration datastore state information is preserved.

```
<?xml version="1.0" encoding="utf-8"?>
 <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
   <cisco-ia:reset xmlns:cisco-ia
     <cisco-ia:reinitialize>true</cisco-ia:reinitialize>
   </cisco-ia:reset>
 </rpc>
```

Note: Some Cisco platforms or Cisco IOS software versions cannot support all of the given functionality at this time. For example, when you send the previous reset to a Catalyst 3850 running IOS 16.3.3, “Reset not supported” error is returned by the Catalyst 3850 to the Centralized Management Platform (laptop) as an RPC reply.

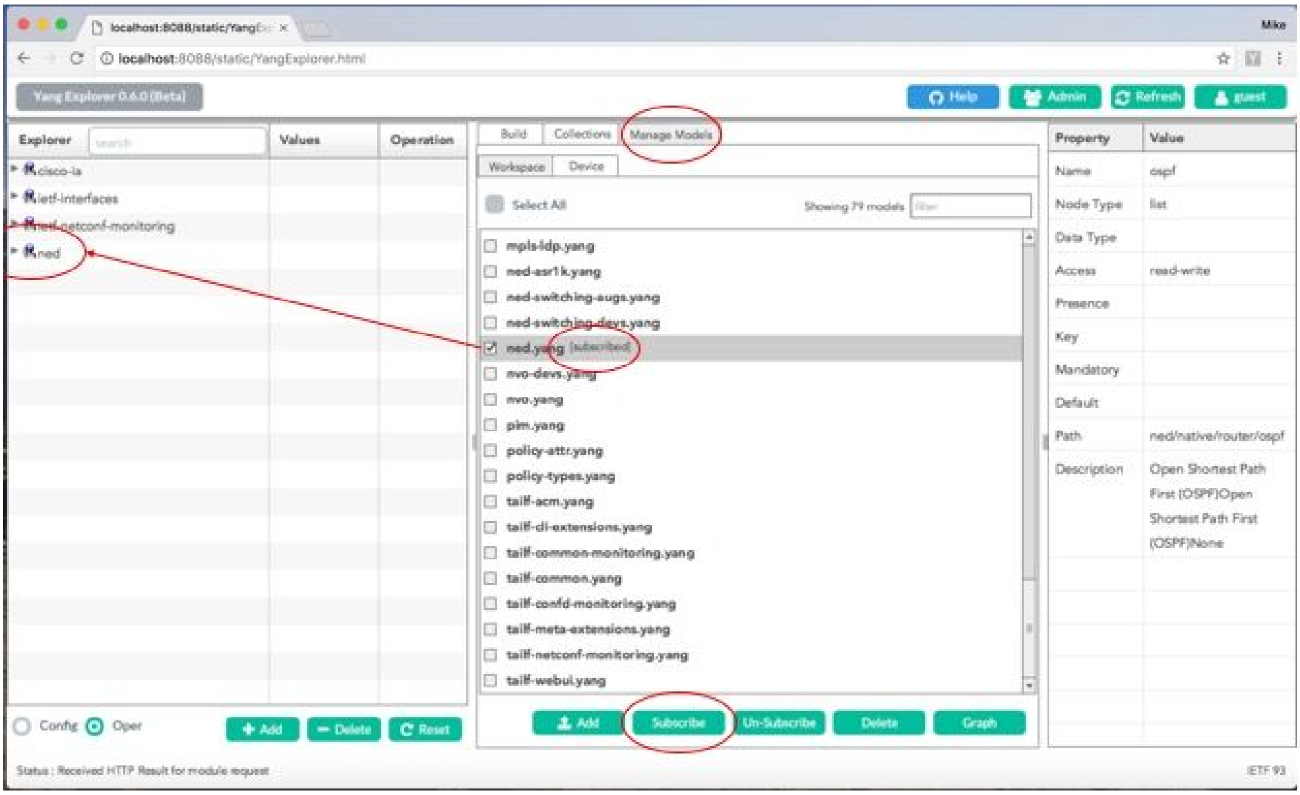
```
<nc:rpc-error xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
   <nc:error-type>application</nc:error-type>
 <nc:error-tag>operation-failed</nc:error-tag>
```

```
    <nc:error-severity>error</nc:error-severity>
    <nc:error-path xmlns:cisco-ia
    <nc:error-message lang="en" xmlns="https://www.w3.org/XML/1998/namespace">Reset not supported</nc:er
    <nc:error-info>
      <nc:bad-element>reset</nc:bad-element>
    </nc:error-info>
  </nc:rpc-error>
```

ned.yang Data Model

The Network Elements Driver (NED) data models such as ned.yang provide the most power in terms of the Cisco device (Catalyst 3850) configuration. Here are some screencaptures that demonstrate this.

The first step is to **Subscribe** to the ned.yang data model so that it appears in the Explorer section on the left of the YANG Explorer application GUI.



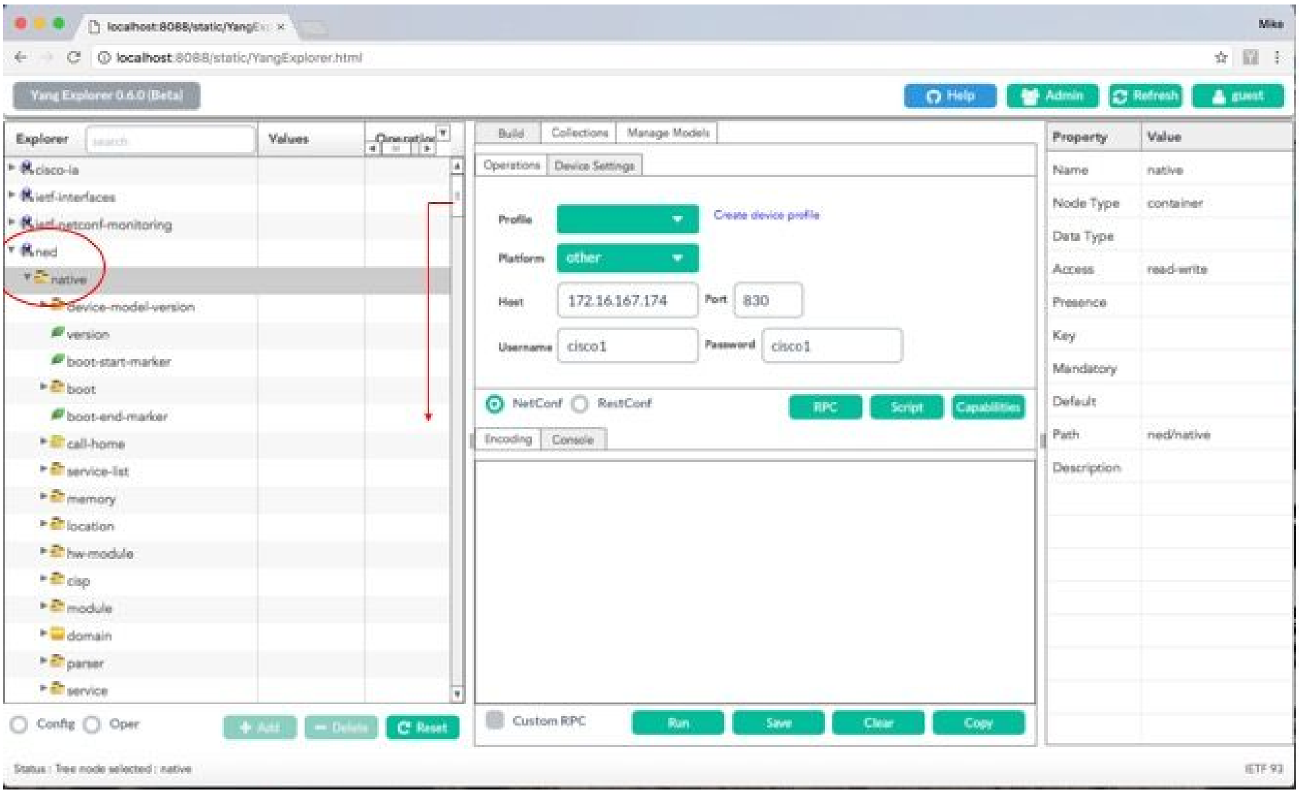
Showing 79 models

□ mpls-ldp.yang
□ ned-asr1k.yang
□ ned-switching-augs.yang
□ ned-switching-devs.yang
☑ ned.yang (subscribed)
□ nvo-devs.yang
□ nvo.yang
□ pim.yang
□ policy-attr.yang
□ policy-types.yang
□ tailf-acm.yang
□ tailf-cli-extensions.yang
□ tailf-common-monitoring.yang
□ tailf-common.yang
□ tailf-confd-monitoring.yang
□ tailf-meta-extensions.yang
□ tailf-netconf-monitoring.yang
□ tailf-webui.yang

Property	Value
Name	ospf
Node Type	list
Data Type	
Access	read-write
Presence	
Key	
Mandatory	
Default	
Path	ned/native/router/ospf
Description	Open Shortest Path First (OSPF)Open Shortest Path First (OSPF)None

Status : Received HTTP Result for module request

Scrolling through the available options in the Explorer section on the left side of the YANG Explorer application, GUI shows a long list of configurable Catalyst 3850 features in the ned.yang data model.

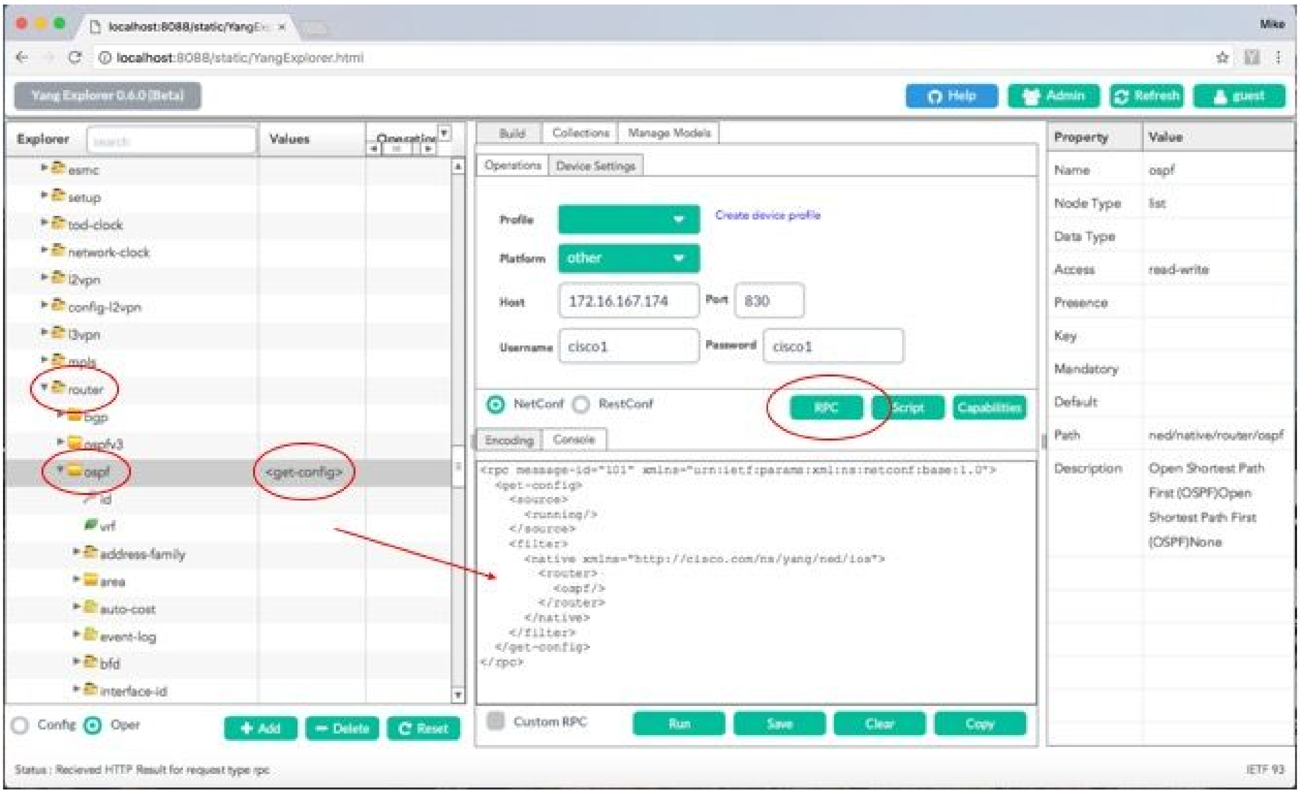


Profile:
Platform: other
Host: 172.16.167.174 Port: 830
Username: cisco1 Password: cisco1
● NetConf ○ RestConf

Property	Value
Name	native
Node Type	container
Data Type	
Access	read-write
Presence	
Key	
Mandatory	
Default	
Path	ned/native
Description	

Status : Tree node selected : native

As an example, these screencaptures demonstrate how to display the OSPF routing configuration of the Catalyst 3850 after first scrolling down the list of available ned.yang data model configuration options in the Explorer section on the left side of the YANG Explorer application GUI. The ospf sub-option is located inside of the router option. The associated get-config RPC is generated when you select the **RPC** button.



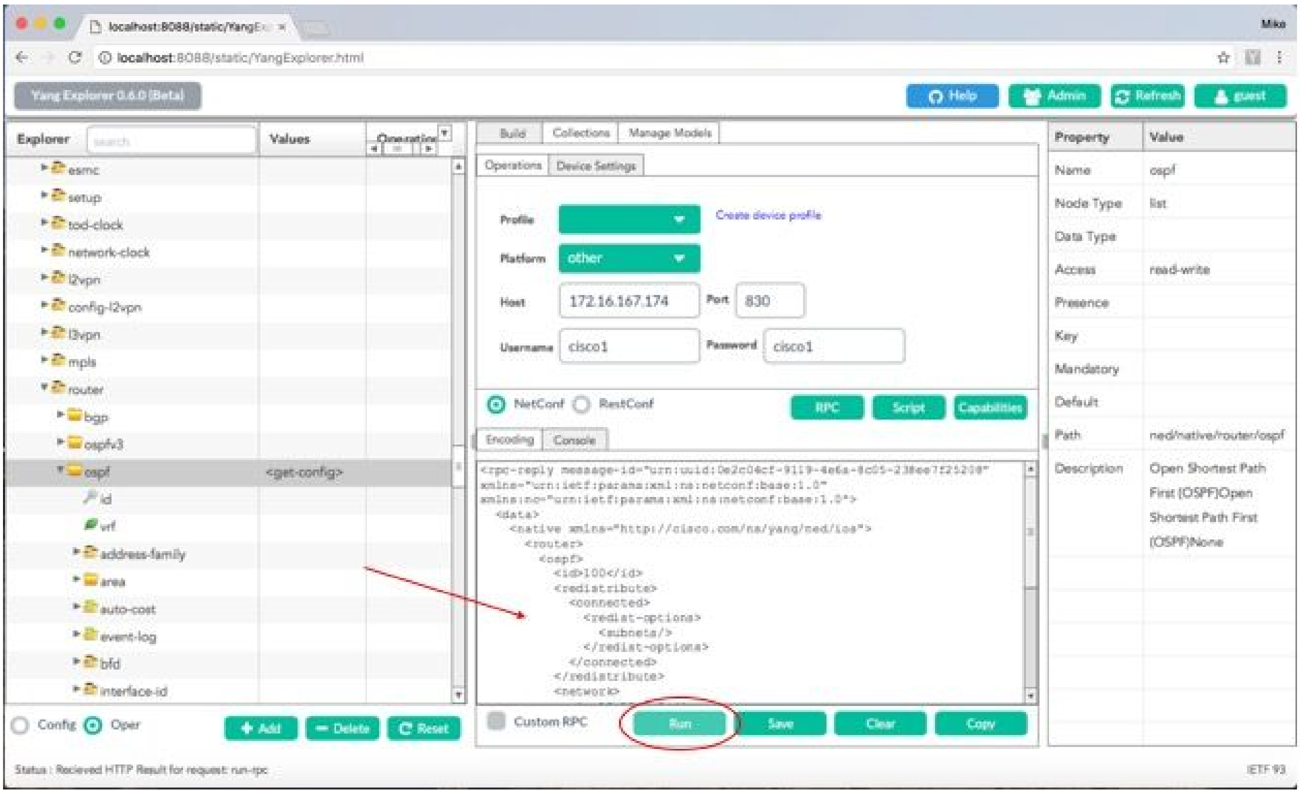
Profile:
Platform: other
Host: 172.16.167.174 Port: 830
Username: cisco1 Password: cisco1
● NetConf ○ RestConf

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

Property	Value
Name	ospf
Node Type	list
Data Type	
Access	read-write
Presence	
Key	
Mandatory	
Default	
Path	ned/native/router/ospf
Description	Open Shortest Path First (OSPF)Open Shortest Path First (OSPF)None

Status : Recieved HTTP Result for request type rpc

Next, **Run** is selected in order to send the RPC message to the Catalyst 3850 via NETCONF. The Catalyst 3850 replies back with its OSPF routing configuration.



ospf <get-config>

○ Config ● Oper

Profile

Platform other

Host 172.16.167.174 Port 830

Username cisco1 Password cisco1

● NetConf ○ RestConf

```
<rpc-reply message-id="urn:uuid:0e2c04cf-9119-4e6a-8c05-238ee7f25208"
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/ned/ios">
      <router>
        <ospf>
          <id>100</id>
          <redistribute>
            <connected>
              <redist-options>
                <subnets/>
              </redist-options>
            </connected>
          </redistribute>
          <network>
```

Property	Value
Name	ospf
Node Type	list
Data Type	
Access	read-write
Presence	
Key	
Mandatory	
Default	
Path	ned/native/router/ospf
Description	Open Shortest Path First (OSPF)Open Shortest Path First (OSPF)None

Status : Recieved HTTP Result for request: run-rpc

Here is an expansion of the OSPF routing configuration returned by the Catalyst 3850 in response to the get-config RPC operation.

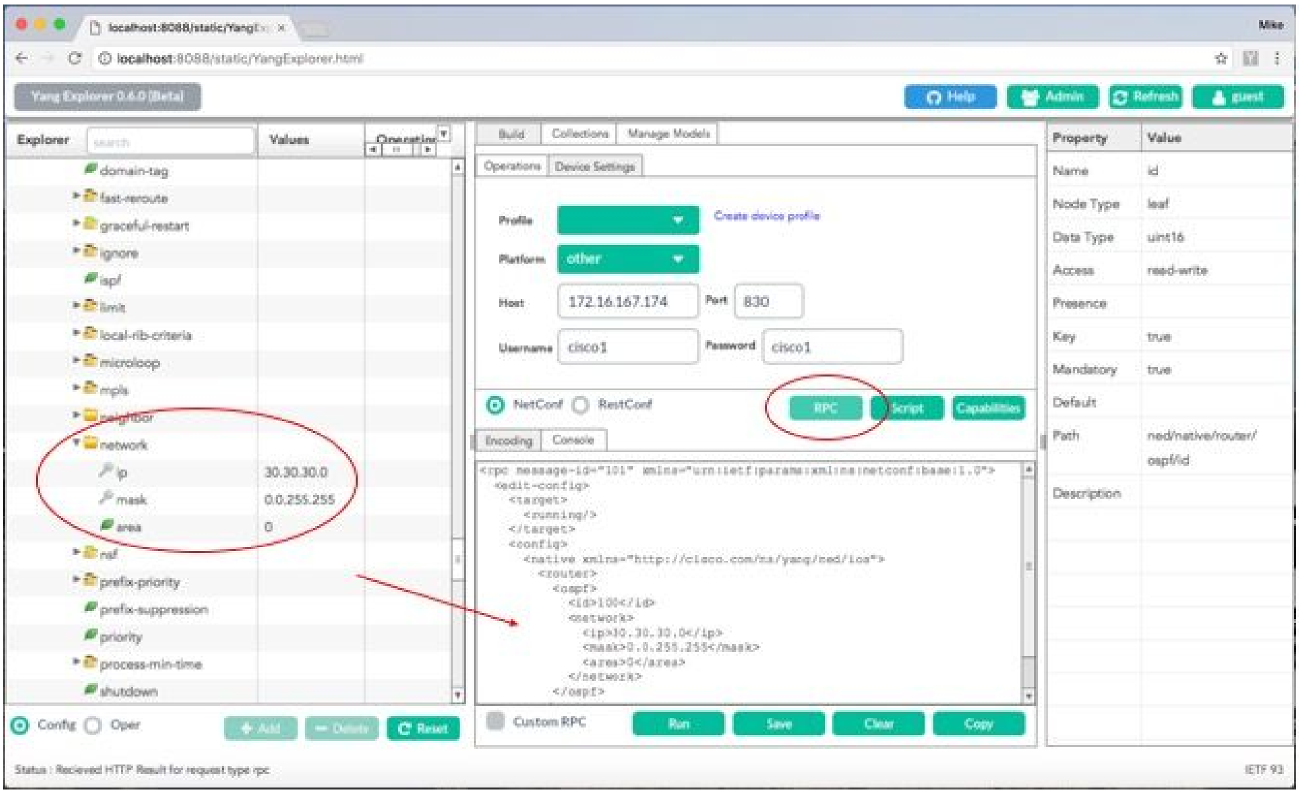
```
<rpc-reply message-id="urn:uuid:0e2c04cf-9119-4e6a-8c05-238ee7f25208" xmlns="urn:ietf:params:xml:ns:net
  <data>
    <native xmlns>
      <router>
        <ospf>
          <id>100</id>
          <redistribute>
            <connected>
              <redist-options>
                <subnets/>
              </redist-options>
            </connected>
          </redistribute>
          <network>
            <ip>10.10.0.0</ip>
            <mask>0.0.255.255</mask>
            <area>0</area>
          </network>
          <network>
            <ip>10.20.0.0</ip>
            <mask>0.0.255.255</mask>
            <area>0</area>
          </network>
          <network>
            <ip>10.100.0.0</ip>
            <mask>0.0.255.255</mask>
            <area>0</area>
          </network>
        </ospf>
      </router>
    </native>
  </data>
</rpc-reply>
```

The YANG formatted OSPF routing configuration that was retrieved from the Catalyst 3850 via NETCONF is human readable and matches what is seen when you look at the Catalyst 3850 configuration via the Catalyst 3850's CLI.

<#root>

```
3850-1#

show running-config | section ospf


 router ospf 100
 redistribute connected subnets
 network 10.10.0.0 0.0.255.255 area 0
 network 10.20.0.0 0.0.255.255 area 0
 network 10.100.0.0 0.0.255.255 area 0
 3850-1#
```

If desired, the ned.yang data model can also be used to modify the OSPF routing configuration. In this example, new network parameters are added to the existing OSPF routing configuration on the Catalyst 3850 by first entering the desired parameters in the Explorer section of the Yang Explorer application GUI on the left (OSPF router ID 100 was also input but is not seen due to Explorer screen scrolling) and then generating the associated YANG formatted RPC and hit the **RPC** button.



Explorer	Values
ip	30.30.30.0
mask	0.0.255.255
area	0

Profile

Platform: other

Host: 172.16.167.174 Port: 830

Username: cisco1 Password: cisco1

● NetConf ○ RestConf

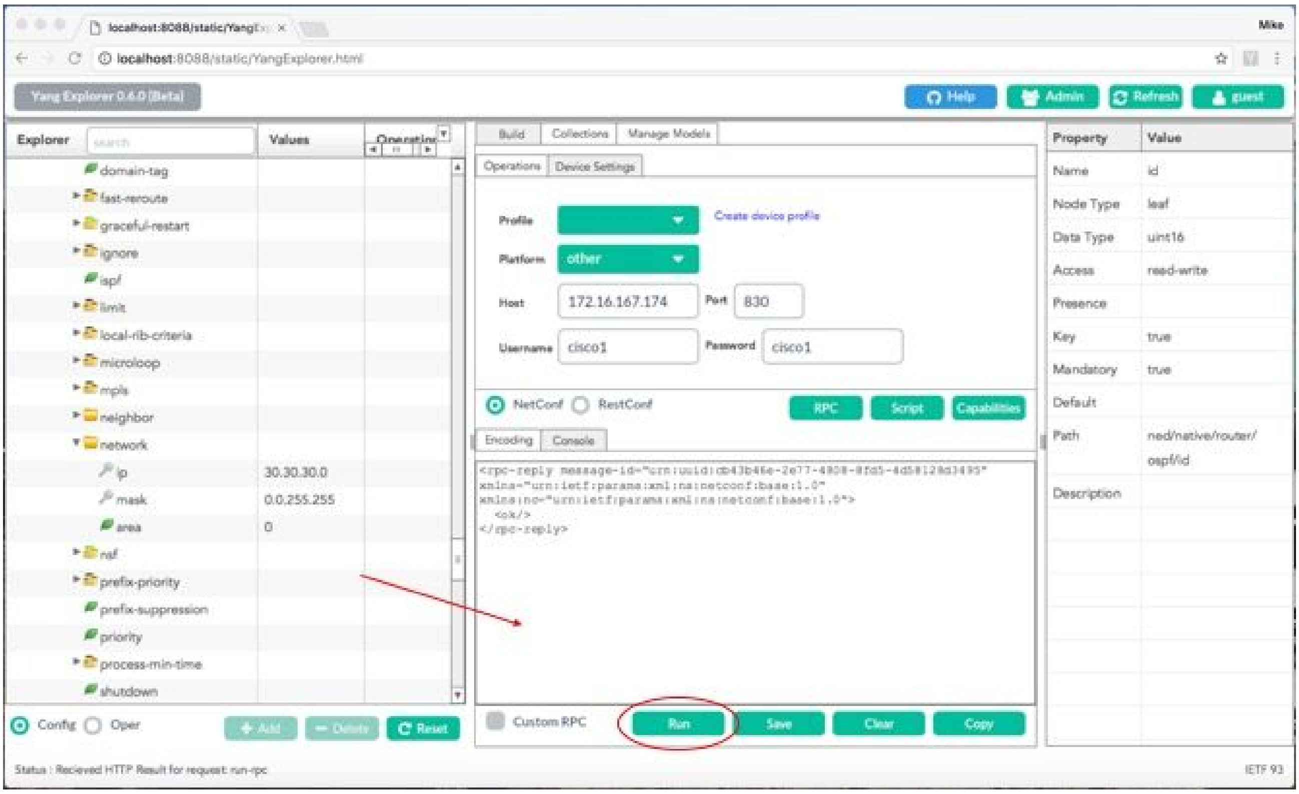
```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <native xmlns="http://cisco.com/ns/yang/ned/ios">
        <router>
          <ospf>
            <id>100</id>
            <network>
              <ip>30.30.30.0</ip>
              <mask>0.0.255.255</mask>
              <area>0</area>
            </network>
          </ospf>
```

Property	Value
Name	id
Node Type	leaf
Data Type	uint16
Access	read-write
Presence	
Key	true
Mandatory	true
Default	
Path	ned/native/router/ ospf/id
Description	

● Config ○ Oper

Status : Recieved HTTP Result for request type rpc

Next, **Run** is selected in order to send the RPC message to the Catalyst 3850 via NETCONF. The Catalyst 3850 replies back with an ok message to let the user know the operation was successful.



ip	30.30.30.0
mask	0.0.255.255
area	0

Profile:
Platform: other
Host: 172.16.167.174 Port: 830
Username: cisco1 Password: cisco1
● NetConf ○ RestConf

```
<rpc-reply message-id="urn:uuid:db43b46e-2e77-4808-8fd5-4d58128d3495"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

Property	Value
Name	id
Node Type	leaf
Data Type	uint16
Access	read-write
Presence	
Key	true
Mandatory	true
Default	
Path	ned/native/router/ ospf/id
Description	

Status : Recieved HTTP Result for request: run-rpc

This NETCONF/YANG RPC operation to modify the OSPF routing configuration via the ned.yang data model is reflected in the Catalyst 3850 configuration as seen via the Catalyst 3850's CLI. There is also a syslog message on the Catalyst 3850 that indicates that a configuration change was made via NETCONF.

```
3850-1#

*Jan 30 14:13:41.659: %DMI-5-CONFIG_I:Switch 1 R0/0: nesd: Configured from NETCONF/RESTCONF by cisco1,

3850-1# show running-config | section ospf
 router ospf 100
 redistribute connected subnets
 network 10.10.0.0 0.0.255.255 area 0
 network 10.20.0.0 0.0.255.255 area 0
 network 10.30.0.0 0.0.255.255 area 0 ------> new line added to OSPF configuration
 network 10.100.0.0 0.0.255.255 area 0
 3850-1#
```

Refer to the save-config operation mentioned in the previous section cisco-ia.yang Data Model for details on how to save the running-config to the startup-config on the Catalyst 3850 via NETCONF/YANG.

Python Scripting

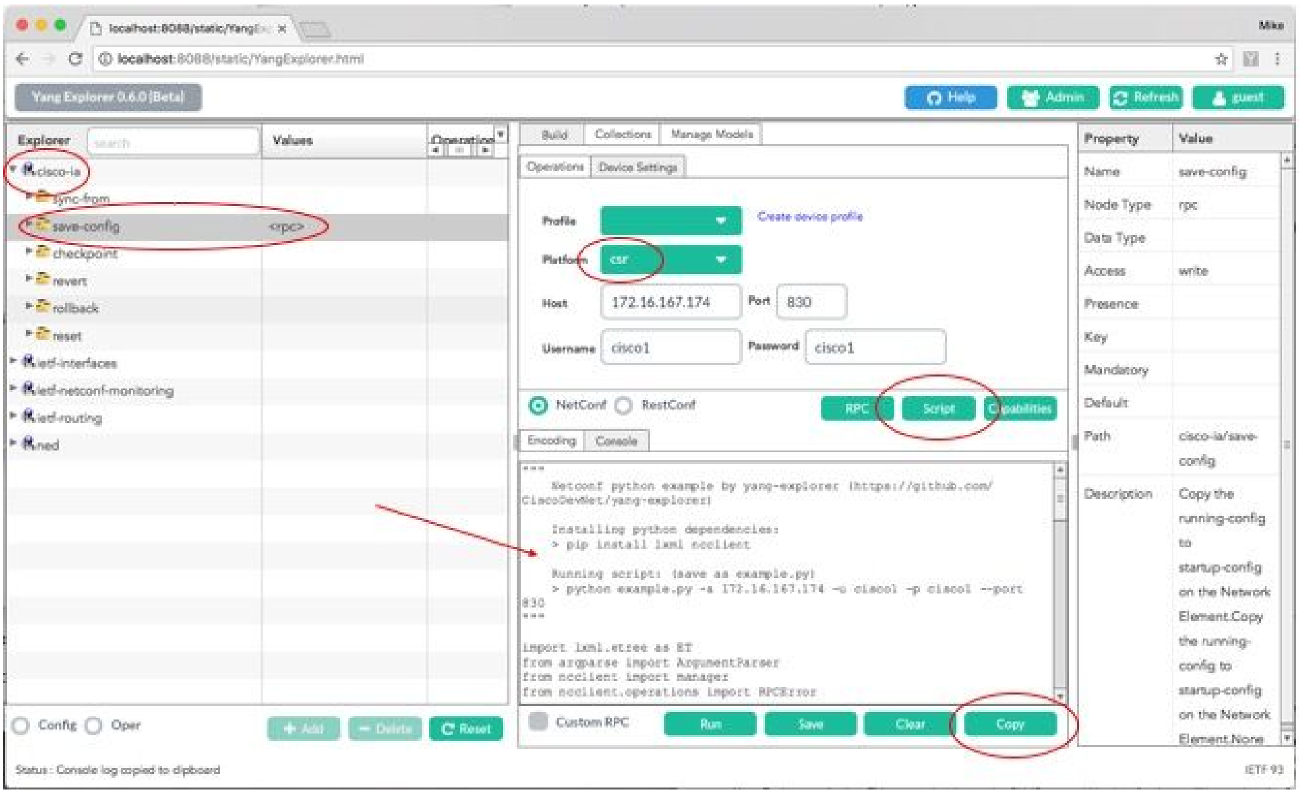
Generate a Python Script from the Yang Explorer Application GUI

The Yang Explore application GUI can also be used to generate a Python script for a given NETCONF/YANG operation. A key benefit of Python scripting is that it allows for the orchestration and automation of NETCONF/YANG operations.

In this example, a save-config operation is selected in the Explorer window on the left hand side of the Yang

Explorer application GUI on the centralized management platform (laptop). Next, the **Script** button is selected to generate the Python script. The **Copy** button can then be selected to copy the script so that it can in turn be pasted into a file that can be saved on the centralized management platform (laptop) with a Python .py file extention. For this example, (not shown) this file has been named example.py.

Note: In the next example that uses Platform type other in the GUI resulted in an error when running the Python script. As a result, the"Platform type was changed to csr since the Cisco CSR router also runs Cisco IOS XE software just as the Catalyst 3850 does. This avoided the error.



Profile
Platform: csr
Host: 172.16.167.174 Port: 830
Username: cisco1 Password: cisco1
● NetConf ○ RestConf

```
"""
    Netconf python example by yang-explorer (https://github.com/
CiscoDevNet/yang-explorer)

    Installing python dependencies:
    > pip install lxml ncclient

    Running script: (save as example.py)
    > python example.py -a 172.16.167.174 -u cisco1 -p cisco1 --port
830
"""

import lxml.etree as ET
from argparse import ArgumentParser
from ncclient import manager
from ncclient.operations import RPCError
```

Property	Value
Name	save-config
Node Type	rpc
Data Type	
Access	write
Presence	
Key	
Mandatory	
Default	
Path	cisco-ia/save-config
Description	Copy the running-config to startup-config on the Network Element.Copy the running-config to startup-config on the Network Element.None

Status : Console log copied to clipboard

Here is an expansion of the Python script that was generated and then copy and pasted into a file called **example.py** on the centralized management platform (laptop).

Note: The comments at the start of the example.py file that was generated by the Yang Explorer application GUI include the steps required to run the Python script. The payload includes the NETCONF/YANG operation that the script can execute. In this example it is a save-config operation.

```
""" 
   Netconf python example by yang-explorer (https://github.com/CiscoDevNet/yang-explorer)

   Installing python dependencies: 
   > pip install lxml ncclient

   Running script: (save as example.py) 
   > python example.py -a 172.16.167.174 -u cisco1 -p cisco1 --port 830 
"""

import lxml.etree as ET 
from argparse import ArgumentParser 
from ncclient import manager
from ncclient.operations import RPCError
```

```
payload = """ □<save-config xmlns
□"""

if __name__ == '__main__':
    parser = ArgumentParser(description='Usage:')
    # script arguments □
    parser.add_argument('-a', '--host', type=str, required=True, □
               help="Device IP address or Hostname") □
    parser.add_argument('-u', '--username', type=str, required=True, □
               help="Device Username (netconf agent username)") □
    parser.add_argument('-p', '--password', type=str, required=True, □
               help="Device Password (netconf agent password)") □
    parser.add_argument('--port', type=int, default=830, □
               help="Netconf agent port") □
    args = parser.parse_args()

    # connect to netconf agent □
    with manager.connect(host=args.host, □
                port=args.port, □
                username=args.username, □
                password=args.password, □
                timeout=90, □
                hostkey_verify=False, □
                device_params={'name': 'csr'}) as m:

# execute netconf operation
□try: □
   response = m.dispatch(ET.fromstring(payload)).xml □
   data = ET.fromstring(response) □
except RPCError as e: □
   data = e._raw
# beautify output
□print(ET.tostring(data, pretty_print=True))
```

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

Here is the Catalyst 3850 CLI check before you run the Python script example.py that can save the running-config to the startup-config. At this point, the **shutdown** command is in the running-config but not in the startup-config for interface GigabitEthernet1/0/10.

```
3850-1# show running-config interface gigabitEthernet 1/0/10
 Building configuration...

Current configuration : 49 bytes
 !
 interface GigabitEthernet1/0/10
 shutdown
 end

3850-1# show startup-config | begin 1/0/10
 interface GigabitEthernet1/0/10
 !
 interface GigabitEthernet1/0/11
 !
 interface GigabitEthernet1/0/12
 !
 interface GigabitEthernet1/0/13
 !
```

From a regular terminal prompt on the centralized management platform (laptop), the Python file example.py that was generated by the Yang Explorer application GUI is first copied to the yang-explore directory on the laptop.

```
USER1-M-902T:~ USER1$ pwd
 /Users/USER1
 USER1-M-902T:~ USER1$ cp /Users/USER1/Desktop/example.py /Users/USER1/yang-explorer
 USER1-M-902T:~ USER1$ cd yang-explorer
 USER1-M-902T:yang-explorer USER1$ ls -l
 total 112
 -rw-r--r-- 1 USER1 staff 11358 Jan 4 17:59 LICENSE
 -rw-r--r-- 1 USER1 staff 13635 Jan 4 17:59 README.md
 drwxr-xr-x 12 USER1 staff 408 Jan 4 17:59 YangExplorer
 drwxr-xr-x 7 USER1 staff 238 Jan 4 17:59 default-models
 drwxr-xr-x 3 USER1 staff 102 Jan 4 17:59 docs
 -rw-r--r-- 1 USER1 staff 72 Jan 4 17:59 env.sh
 -rw-r--r--@ 1 USER1 staff 1990 Jan 30 17:50 example.py
 -rw-r--r-- 1 USER1 staff 207 Jan 4 17:59 requirements.txt
 drwxr-xr-x 11 USER1 staff 374 Jan 5 14:37 server
 -rwxr-xr-x 1 USER1 staff 4038 Jan 4 17:59 setup.sh
 -rwxr-xr-x 1 USER1 staff 640 Jan 4 17:59 start.sh
 drwxr-xr-x 5 USER1 staff 170 Jan 4 18:00 v
 USER1-M-902T:yang-explorer USER1$
```

Next, from a regular terminal prompt on the centralized management platform (laptop), these two commands are executed which were provided in the comment section at the start of the example.py file that was generated by the Yang Explorer application GUI (refer to the previous section, Generating a Python Script from the Yang Explorer Application GUI).

```
 USER1-M-902T:yang-explorer USER1$ pip install lxml ncclient
 Collecting lxml
 Downloading lxml-3.7.2.tar.gz (3.8MB)
 100% |████████████████████████████████| 3.8MB 328kB/s
 Collecting ncclient
 Downloading ncclient-0.5.3.tar.gz (63kB)
 100% |████████████████████████████████| 71kB 3.5MB/s
 Requirement already satisfied: setuptools>0.6 in /Library/Frameworks/Python.framework/Versions/2.7/lib
 Collecting paramiko>=1.15.0 (from ncclient)
 Downloading paramiko-2.1.1-py2.py3-none-any.whl (172kB)
 100% |████████████████████████████████| 174kB 3.1MB/s
 Collecting six (from ncclient)
 Using cached six-1.10.0-py2.py3-none-any.whl
 Collecting cryptography>=1.1 (from paramiko>=1.15.0->ncclient)
 Using cached cryptography-1.7.2-cp27-cp27m-macosx_10_6_intel.whl
 Collecting pyasn1>=0.1.7 (from paramiko>=1.15.0->ncclient)
 Using cached pyasn1-0.1.9-py2.py3-none-any.whl
 Collecting cffi>=1.4.1 (from cryptography>=1.1->paramiko>=1.15.0->ncclient)
 Using cached cffi-1.9.1-cp27-cp27m-macosx_10_10_intel.whl
 Collecting enum34 (from cryptography>=1.1->paramiko>=1.15.0->ncclient)
 Using cached enum34-1.1.6-py2-none-any.whl
 Collecting ipaddress (from cryptography>=1.1->paramiko>=1.15.0->ncclient)
 Using cached ipaddress-1.0.18-py2-none-any.whl
 Collecting idna>=2.0 (from cryptography>=1.1->paramiko>=1.15.0->ncclient)
 Using cached idna-2.2-py2.py3-none-any.whl
 Collecting pycparser (from cffi>=1.4.1->cryptography>=1.1->paramiko>=1.15.0->ncclient)
 Downloading pycparser-2.17.tar.gz (231kB)
 100% |████████████████████████████████| 235kB 2.6MB/s
 Installing collected packages: lxml, six, pycparser, cffi, pyasn1, enum34, ipaddress, idna, cryptograp
 Running setup.py install for lxml ... -
 done
 Running setup.py install for pycparser ... done
 Running setup.py install for ncclient ... done
 Successfully installed cffi-1.9.1 cryptography-1.7.2 enum34-1.1.6 idna-2.2 ipaddress-1.0.18 lxml-3.7.2
 USER1-M-902T:yang-explorer USER1$
```

The 2nd command runs the Python script example.py against the Catalyst 3850 at IP address 172.16.167.174 with the username/password cisco1/cisco1 via TCP port 830 (netconf-ssh). The Catalyst 3850 sends an RPC reply back to the centralized management platform (laptop) that the save-config operation was successful.

```
USER1-M-902T:yang-explorer USER1$ python example.py -a 172.16.167.174 -u cisco1 -p cisco1 --port 830

<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:bas
 </result>
 </rpc-reply>

USER1-M-902T:yang-explorer USER1
```

Here is the Catalyst 3850 CLI check after you run the Python script example.py that saved the running-config to the start-up config. The **shutdown** command is now present in both the running-config and the startup-config for interface GigabitEthernet1/0/10 due to the successful save-config NETCONF/YANG operation.

```
3850-1# show running-config interface gigabitEthernet 1/0/10
 Building configuration...

Current configuration : 49 bytes
 !
 interface GigabitEthernet1/0/10
 shutdown
 end

3850-1# show startup-config | begin 1/0/10
 interface GigabitEthernet1/0/10
 shutdown
 !
 interface GigabitEthernet1/0/11
 !
 interface GigabitEthernet1/0/12
 !
 interface GigabitEthernet1/0/13
 !
```

Troubleshoot

This section provides information you can use to troubleshoot your configuration.

NETCONF Error Messages

The NETCONF protocol defines a set of operations and messages that are exchanged between the NETCONF Client (Centralized Management Platform (laptop)) and the NETCONF implementation on the Server device (Catalyst 3850). Commonly used NETCONF operations include:

<get>, <get-config>, <edit-config>, and <rpc>

The format and other constraints on the NETCONF message content are defined by YANG data models. The NETCONF Client and Server interact by sending RPCs.

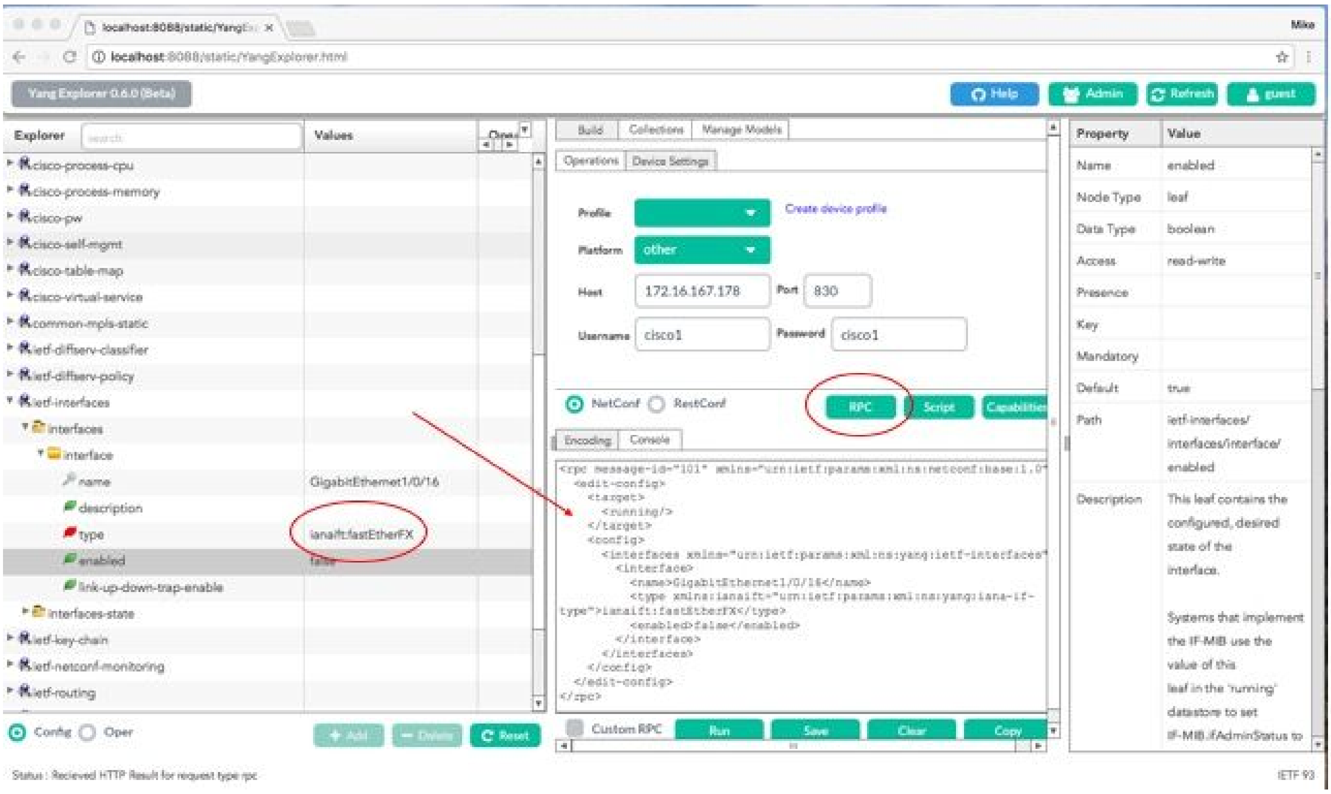
If there is an error in the format of NETCONF message, or the content of the message does not match the definitions in the YANG data models implemented by the device, the NETCONF server on the device can return an RPC error.

```
<error-type>application</error-type>
```

These RPC errors do not indicate that the NETCONF interface is not working, these errors indicate that the client is trying to perform an operation that is not supported by the YANG data models implemented on the server device. Users must review the YANG data models implemented on the server device to identify and resolve the causes for these errors.

RPC Error Example

In this example an incorrect Interface type ianaift:fastEtherFX is used to generate the YANG formatted <edit-config> NETCONF RPC message to send via NETCONF to the Catalyst 3850.



name	GigabitEthernet1/0/16
type	ianaift:fastEtherFX
enabled	false

Profile
Platform: other
Host: 172.16.167.178 Port: 830
Username: cisco1 Password: cisco1

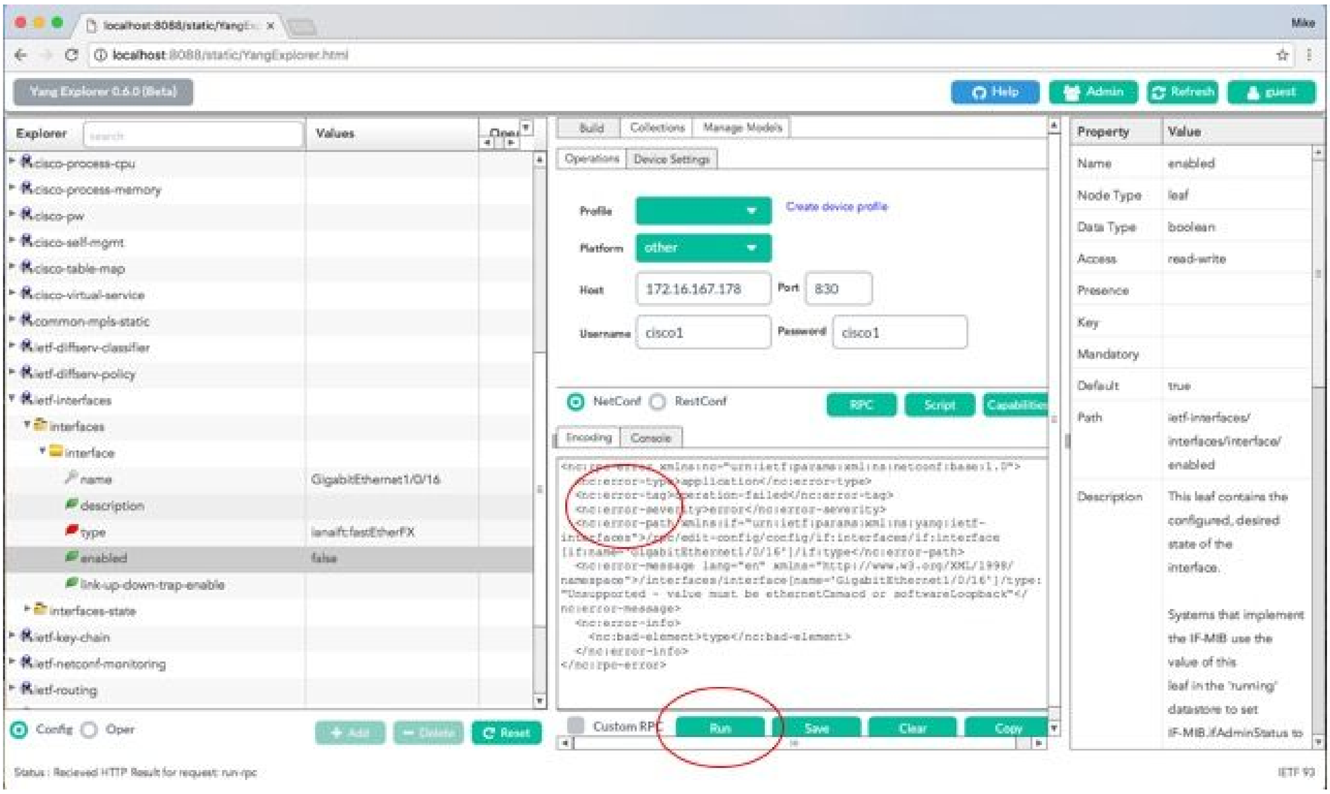
◉ NetConf ○ RestConf

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
        <interface>
          <name>GigabitEthernet1/0/16</name>
          <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-
type">ianaift:fastEtherFX</type>
          <enabled>false</enabled>
        </interface>
      </interfaces>
    </config>
  </edit-config>
</rpc>
```

Property	Value
Name	enabled
Node Type	leaf
Data Type	boolean
Access	read-write
Presence	
Key	
Mandatory	
Default	true
Path	ietf-interfaces/interfaces/interface/enabled
Description	This leaf contains the configured, desired state of the interface. Systems that implement the IF-MIB use the value of this leaf in the 'running' datastore to set IF-MIB.ifAdminStatus to

Status : Recieved HTTP Result for request type rpc

Once **Run** is selected to send the RPC message to the Catalyst 3850, the Catalyst 3850 replies with an error message.



name	GigabitEthernet1/0/16
type	ianaift:fastEtherFX
enabled	false

Profile
Platform: other
Host: 172.16.167.178 Port: 830
Username: cisco1 Password: cisco1

◉ NetConf ○ RestConf

```
<nc:rpc-error xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nc:error-type>application</nc:error-type>
  <nc:error-tag>operation-failed</nc:error-tag>
  <nc:error-severity>error</nc:error-severity>
  <nc:error-path xmlns:if="urn:ietf:params:xml:ns:yang:ietf-
interfaces">/rpc/edit-config/config/if:interfaces/if:interface
[if:name='GigabitEthernet1/0/16']/if:type</nc:error-path>
  <nc:error-message lang="en" xmlns="http://www.w3.org/XML/1998/
namespace">/interfaces/interface[name='GigabitEthernet1/0/16']/type:
"Unsupported - value must be ethernetCsmacd or softwareLoopback"</
nc:error-message>
  <nc:error-info>
    <nc:bad-element>type</nc:bad-element>
  </nc:error-info>
</nc:rpc-error>
```

Property	Value
Name	enabled
Node Type	leaf
Data Type	boolean
Access	read-write
Presence	
Key	
Mandatory	
Default	true
Path	ietf-interfaces/interfaces/interface/enabled
Description	This leaf contains the configured, desired state of the interface. Systems that implement the IF-MIB use the value of this leaf in the 'running' datastore to set IF-MIB.ifAdminStatus to

Status : Recieved HTTP Result for request: run-rpc

Here is the error that was returned by the Catalyst 3850. Notice that it contains an error tag “operation-failed” and further detail that concerns the error says “Unsupported - value must be ethernetCsmacd or softwareLoopback"</nc:error-message>”.

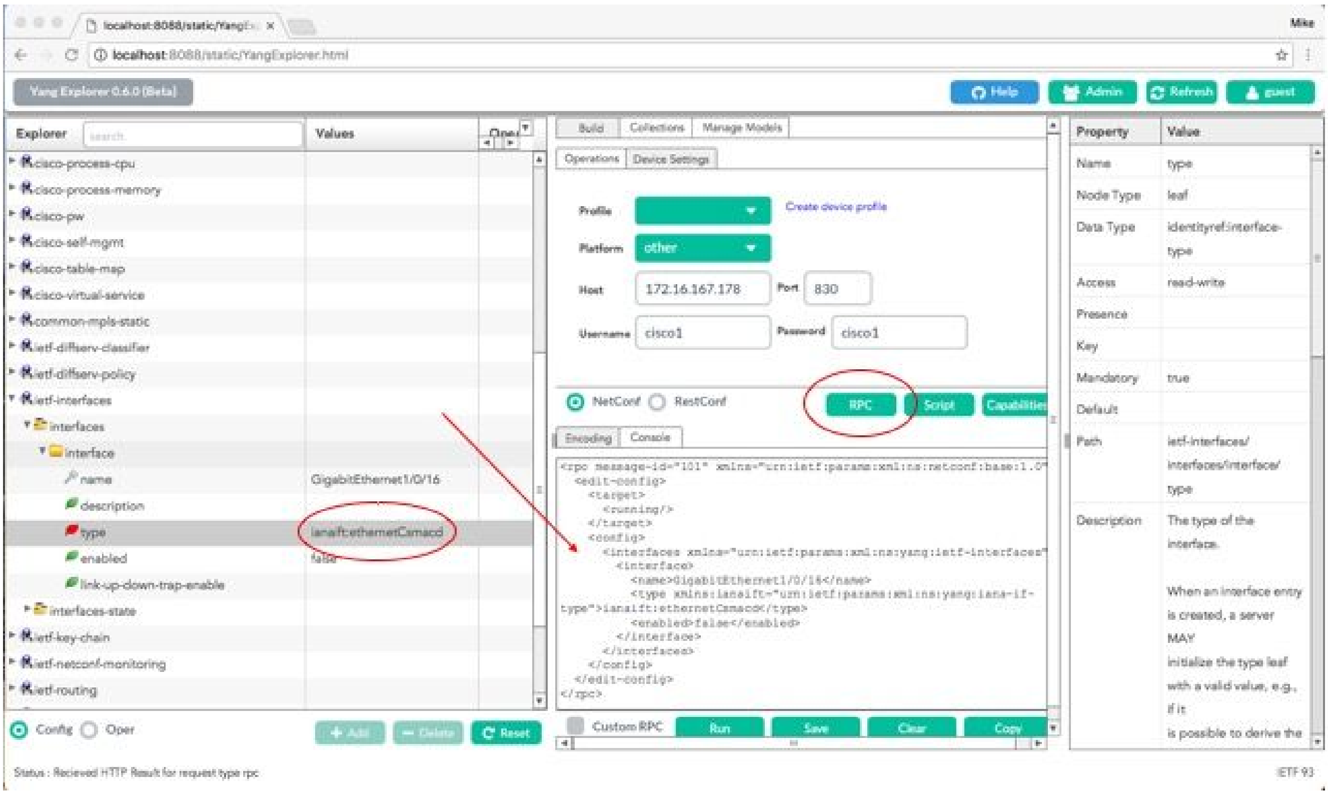
<#root>

```
<nc:rpc-error xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
   <nc:error-type>application</nc:error-type>
  <nc:error-tag>operation-failed</nc:error-tag>
   <nc:error-severity>error</nc:error-severity>
   <nc:error-path xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces">/rpc/edit-config/config/if:int
   <nc:error-message lang="en" xmlns="https://www.w3.org/XML/1998/namespace">/interfaces/interface[name=
```

"Unsupported - value must be ethernetCsmacd or softwareLoopback"</nc:error-message>

```
   <nc:error-info>
     <nc:bad-element>type</nc:bad-element>
   </nc:error-info>
 </nc:rpc-error>
```

Next, let us fix the error and specify the correct interface type ianaift:ethernetCsmacd in The RPC message sent to the Catalyst 3850 so that the Catalyst 3850 replies with an ok message instead of an error.



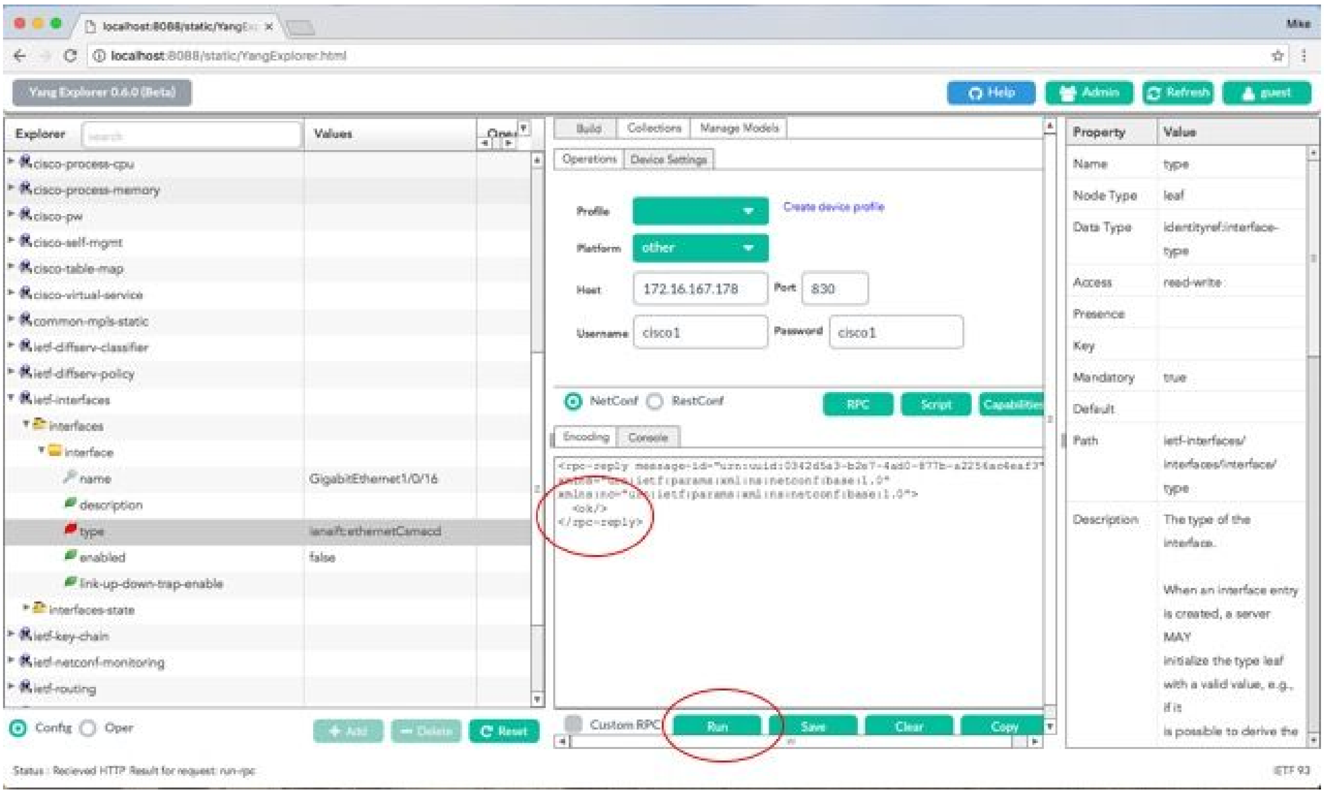
name: GigabitEthernet1/0/16
type: ianaift:ethernetCsmacd
enabled: false

Profile:
Platform: other
Host: 172.16.167.178 Port: 830
Username: cisco1 Password: cisco1
◉ NetConf ○ RestConf

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
        <interface>
          <name>GigabitEthernet1/0/16</name>
          <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-
type">ianaift:ethernetCsmacd</type>
          <enabled>false</enabled>
        </interface>
      </interfaces>
    </config>
  </edit-config>
</rpc>
```

Property	Value
Name	type
Node Type	leaf
Data Type	identityref:interface-type
Access	read-write
Presence	
Key	
Mandatory	true
Default	
Path	ietf-interfaces/interfaces/interface/type
Description	The type of the interface. 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

This time, once **Run** is selected to send the RPC message to the Catalyst 3850, the Catalyst 3850 replies with an ok message to indicate that the operation was successful.



Explorer	Values
name	GigabitEthernet1/0/16
type	ianaift:ethernetCsmacd
enabled	false

Profile:
Platform: other
Host: 172.16.167.178 Port: 830
Username: cisco1 Password: cisco1

● NetConf ○ RestConf

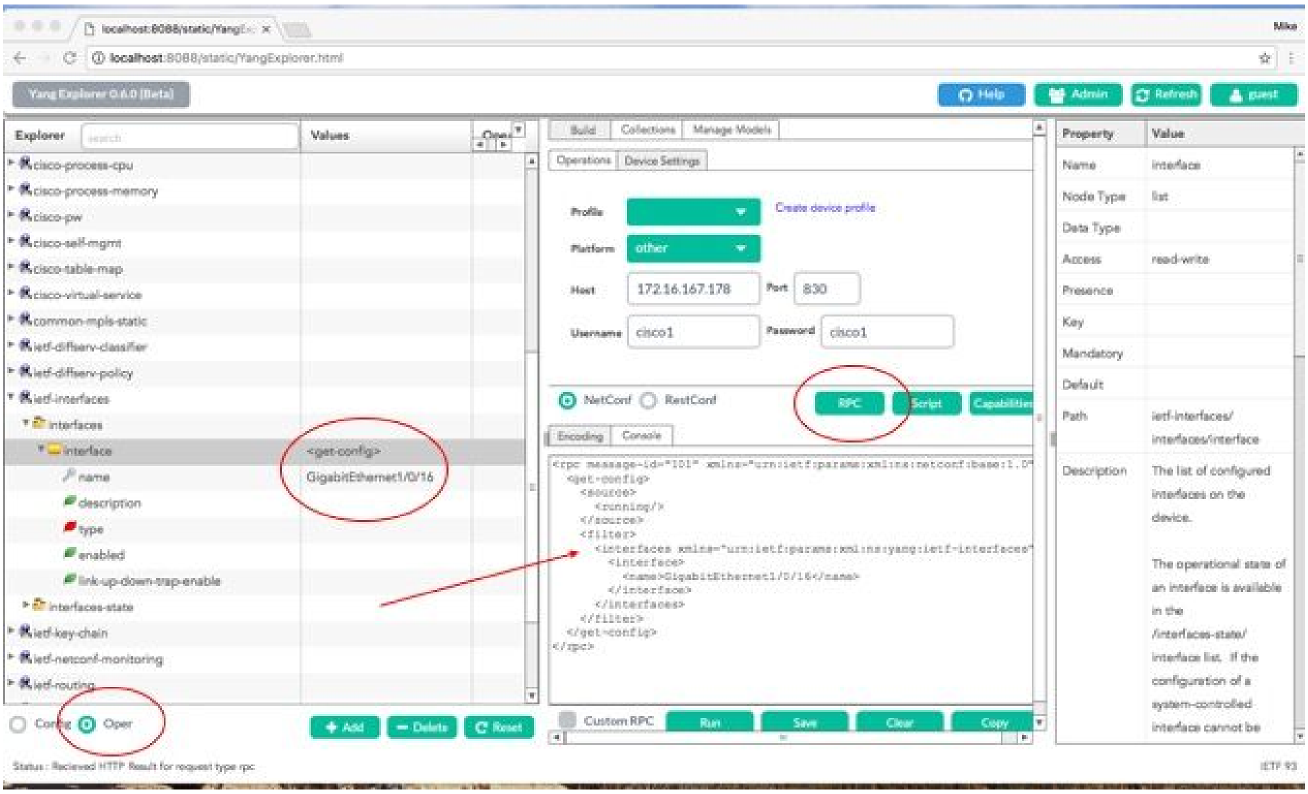
```
<rpc-reply message-id="urn:uuid:0342d5a3-b2e7-4ad0-877b-a2254ac4eaf3"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

● Config ○ Oper

Property	Value
Name	type
Node Type	leaf
Data Type	identityref:interface-type
Access	read-write
Presence	
Key	
Mandatory	true
Default	
Path	ietf-interfaces/interfaces/interface/type
Description	The type of the interface. 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

Status : Recieved HTTP Result for request: run-rpc

Tip: When unsure what the correct Explorer Values format can be, the configuration that exists can be looked at before you attempt to make a change to it's parameters. This can be done with the **get-config** operation (Oper) as shown.



Explorer	Values
interface	<get-config>
name	GigabitEthernet1/0/16

Profile:
Platform: other
Host: 172.16.167.178 Port: 830
Username: cisco1 Password: cisco1

● NetConf ○ RestConf

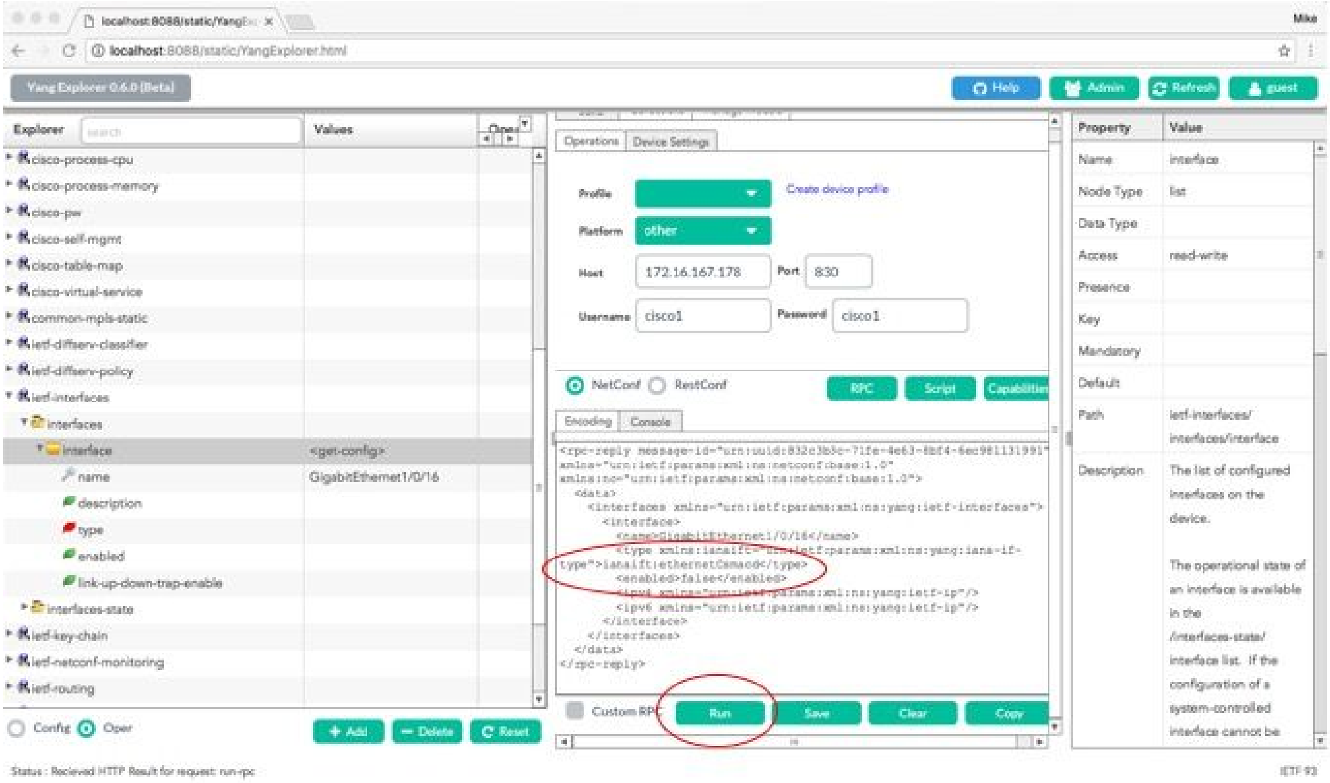
```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
        <interface>
          <name>GigabitEthernet1/0/16</name>
        </interface>
      </interfaces>
    </filter>
  </get-config>
</rpc>
```

○ Config ● Oper

Property	Value
Name	interface
Node Type	list
Data Type	
Access	read-write
Presence	
Key	
Mandatory	
Default	
Path	ietf-interfaces/interfaces/interface
Description	The list of configured interfaces on the device. The operational state of an interface is available in the /interfaces-state/interface list. If the configuration of a system-controlled interface cannot be

Status : Recieved HTTP Result for request type rpc

Once **Run** is selected to send the RPC message to the Catalyst 3850, the Catalyst 3850 replies with the YANG formatted interface configuration which shows that interface type is ianaift:ethernetCsmacd.



interface <get-config>

name GigabitEthernet1/0/16

Profile:

Platform: other

Host: 172.16.167.178 Port: 830

Username: cisco1 Password: cisco1

```
<rpc-reply message-id="urn:uuid:832c3b3c-71fe-4e63-8bf4-6ec981131991"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
      <interface>
        <name>GigabitEthernet1/0/16</name>
        <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-
type">ianaift:ethernetCsmacd</type>
        <enabled>false</enabled>
        <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"/>
        <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"/>
      </interface>
    </interfaces>
  </data>
</rpc-reply>
```

Property	Value
Name	interface
Node Type	list
Data Type	
Access	read-write
Presence	
Key	
Mandatory	
Default	
Path	ietf-interfaces/interfaces/interface
Description	The list of configured interfaces on the device. The operational state of an interface is available in the /interfaces-state/interface list. If the configuration of a system-controlled interface cannot be

Status : Recieved HTTP Result for request: run-rpc

Other RPC Error Type Examples

1. "In-use" (config-locked) RPC Error Reply Message

This is a NETCONF error response to an <edit-config> request. The <error-tag> indicates "in-use”. The response indicates that the server device (Catalyst 3850) NETCONF running datastore is currently locked and the NETCONF <edit-config> operation could not be performed at this time. This does not indicate an error in the NETCONF interface implementation. If a NETCONF client attempts a write to the NETCONF running datastore when the datastore is in use, the client receives this RPC response. The NETCONF client can retry the NETCONF edit-config message. This response can be received when the device is performing a sync-from-device internal operation to synchronize the NETCONF running datastore with the device IOSd configuration.

NETCONF Response from server (Catalyst 3850) to client (Centralized Management Platform (Laptop)).

```
<?xml version="1.0" encoding="utf-8"?>
 <rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="3">
   <rpc-error>
     <error-type>application</error-type>
     <error-tag>in-use</error-tag>
     <error-severity>error</error-severity>
     <error-app-tag>config-locked</error-app-tag>
     <error-info>
       <session-id>0</session-id>
     </error-info>
   </rpc-error>
 </rpc-reply>
```

2. "Data-missing" RPC Error Reply Message

In this example, an <edit-config> RPC was sent to the Catalyst 3850 for a loopback interface that was not configured. An error was returned since you cannot configure an interface that does not exist on the Catalyst 3850.

```
<?xml version="1.0" encoding="utf-8"?>
 <rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="3">
   <rpc-error>
     <error-type>application</error-type>
     <error-tag>data-missing</error-tag>
     <error-severity>error</error-severity>
     <error-path xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces">/rpc/edit-config/config/if:inte
     <error-message xml:lang="en">/interfaces/interface[name='Loopback1111']/type is not configured</er
     <error-info>
       <bad-element>type</bad-element>
     </error-info>
   </rpc-error>
 </rpc-reply>
```

3. "Missing Data Model" RPC Error Reply Message

If a request is made for a data model that does not exist on the Catalyst 3850 or a request is made for a leaf that is not implemented in a data model, the Server (Catalyst 3850) responds with an empty data response. This is expected behavior.

Tip: Use the NETCONF capabilities functionality to determine which data models are supported by the Catalyst software. See section 2. of Configuring the Centralized Management Platform (laptop).

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"/>
```

4. "Invalid-value" RPC Error Reply Message

In some cases a NETCONF message can contain content that is valid based on the YANG data models, however, the device (Catalyst 3850) is unable to implement what is requested. When the NETCONF interface on the Catalyst 3850 sends configurations to IOSd that IOSd can’t successfully apply, a specific RPC error response is returned to the NETCONF Client.

In this example, an invalid logging buffered value of bogus is sent in the RPC message to the Catalyst 3850. The error-tag in the reply from the Catalyst 3850 indicates invalid-value. The error-message indicates that the Catalyst 3850 IOS Parser was unable to configure the logging buffered severity level to bogus since this is not a valid value.

```
<?xml version="1.0" encoding="utf-8"?>
 <rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="6">
   <rpc-error>
     <error-type>application</error-type>
     <error-tag>invalid-value</error-tag>
     <error-severity>error</error-severity>
     <error-message xml:lang="en">inconsistent value: Device refused command "logging buffered bogus" a
```

```
  </rpc-error>
</rpc-reply>
```