# Obtain Data Models

The data models are available in the mgbl pie software package. Installing a package on the router installs specific features that are part of that package. Cisco IOS XR software is divided into various software packages to select the features to run on the router. Each package contains components that perform a specific set of router functions, such as routing, security, and so on.

1. Verify that the data models available are using `netconf-monitoring` request.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
 <get>
   <filter type="subtree">
     <netconf-state xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
     <schemas/>
     </netconf-state>
   </filter>
 </get>
</rpc>
```

All IOS XR and System Admin YANG models are displayed.

The models are in the `.yang` format. A model with:

- `-oper` in the model name indicates an operational model.
- `-cfg` indicates a configuration model.
- `-act` indicates a NETCONF actions model. For example, Cisco-IOS-XR-ipv4-ospf-act.yang is an action model for OSPF.

  The YANG models can be retrieved from the router without logging into the router using **get-schema** command:

Get Schema List (data will be used in step 2).

```
<get>
<filter type="subtree">
<netconf-state xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
<schemas/>
</netconf-state>
</filter>
</get>
</rpc>
```

```
TRACE: 2016/06/13 11:11:42 transport.go:104: Reading from connection
TRACE: 2016/06/13 11:11:42 gnc_main.go:587: Session established (Id: 1009461378)
TRACE: 2016/06/13 11:11:42 session.go:93: Request:
<rpc message-id="16a79f87-1d47-4f7a-a16a-9405e6d865b9"
```

```
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"><get><filter type="subtree"><netconf-state

xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring"><schemas/></netconf-state></filter></get></rpc>
TRACE: 2016/06/13 11:11:42 transport.go:104: Reading from connection
TRACE: 2016/06/13 11:11:42 session.go:117:
Response:
#143589
<rpc-reply message-id="16a79f87-1d47-4f7a-a16a-9405e6d865b9"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<data>
<netconf-state xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
<schemas>
<schema>
<identifier>Cisco-IOS-XR-crypto-sam-oper</identifier>
<version>2015-01-07</version>
<format>yang</format>
<namespace>http://cisco.com/ns/yang/Cisco-IOS-XR-crypto-sam-oper</namespace>
<location>NETCONF</location>
</schema>
<schema>
<identifier>Cisco-IOS-XR-crypto-sam-oper-sub1</identifier>
<version>2015-01-07</version>
<format>yang</format>
<namespace>http://cisco.com/ns/yang/Cisco-IOS-XR-crypto-sam-oper</namespace>
<location>NETCONF</location>
</schema>
<schema>
<identifier>Cisco-IOS-XR-snmp-agent-oper</identifier>
<version>2015-10-08</version>
<format>yang</format>
<namespace>http://cisco.com/ns/yang/Cisco-IOS-XR-snmp-agent-oper</namespace>
<location>NETCONF</location>
</schema>
------------<truncated>--------------
```

All the models on the router are displayed.

**Note** You can register at https://wwwin-github.cisco.com and then refer the following link for SVO Yang models:https://github.com/YangModels/yang/tree/master/vendor/cisco/svo

- Commit Configuration, on page 2

# Commit Configuration

Commit the configuration to set the new values in the current running configuration.

The configuration can also be committed through a `confimed-commit` operation. NETCONF supports confirmed-commit RPC. This RPC requires an explicit confirmation from the user before the configuration takes effect on the router. This feature helps in verifying that the change in the configuration works correctly and does not cause fluctuation in the management connectivity. If the configuration change causes loss of management connectivity, the configuration is automatically rolled back to the previous committed configuration after the default `confirm-timeout` period of 600 seconds.

To commit a configuration, use </commit> RPC:

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
```

```
    <commit/>
</rpc>
```

To confirm-commit a configuration:

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <commit>
        <confirmed/>
    </commit>
</rpc>
```

The confirmed-commit capability supports the <cancel-commit> operation and the <confirmed>, <confirm-timeout>, <persist>, and <persist-id> parameters for the <commit> operation.

```
<rpc message-id="101"
          xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
       <commit>
         <confirmed/>
         <persist>IQ,d4668</persist>
         <confirm-timeout>120</confirm-timeout>
       </commit>
</rpc>
```

A confirmed-commit request will fail with `Datastore Locked` error if:

- Another operation is performed between a confirmed-commit and a confirming-commit operation
- Another session has an active confirmed-commit request and a persist ID was not provided
- A persist ID was provided but did not match the persist ID of the active confirmed-commit session