Enhancements to Data Models

This section provides an overview of the enhancements made to data models.

NETCONF Accounting Logs

Table 1: Feature History Table

Feature Name	Release Information	Description
Accounting Records for NETCONF Operations	Release 7.6.1	Depending on the accounting configuration command you use, every NETCONF operation that the router performs is reported to the local server as syslog messages or remote AAA servers like TACACS+ as accounting messages, or both.

With this feature, you can view the accounting logs of all NETCONF operations such as `edit-config`, `get-config`, `get` operations that are performed on the router. The logs include the following data:

- RPC name
- Commit ID
- Session ID
- Message ID
- XPath

For more information, see *Implementing System Logging* chapter in the *System Monitoring Configuration Guide for Cisco ASR 9000 Series Routers*.

To enable NETCONF accounting logs, do the following steps:

Step 1 Enter the configuration mode.

Example:

```
Router#conf t
```

Step 2 Create a method list for accounting.

Example:

```
Router(config)#aaa accounting commands default start-stop group tacacs+ local
```

Use one or both of the method list value to enable system accounting.

- **TACACS+**—The logs are stored on the TACACS+ server.
- **Local**—The logs are stored in a user-specified file on the router. The maximum file size is 2047 MB.

Step 3 Commit the configuration.

Example:

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

Note Syslog message about start and end of the session with details such as session ID, user, and remote address information is displayed for NETCONF operations only when both the EXEC accounting and local command accounting is enabled.

```
Router(config)#aaa accounting exec default start-stop group tacacs+
Router(config)#aaa accounting commands default start-stop local
```

Example

NETCONF Accounting Logs

With the RPC commit operation, the configuration changes are reported in the form of CLI commands. In this example, the `edit-config` operation is converted into its equivalent CLI `aaa accounting system default start-stop none` command in the logs; the user ID and session ID details are logged.

```
RP/0/RP0/CPU0:Mar 15 17:04:34.950 UTC: locald_DLRSC[233]: %SECURITY-LOCALD-6-LOCAL_CMD_ACCT
 :
 RPC CMD: "aaa accounting system default start-stop none" by <user> from TTY
netconf-3745105668
10.0.0.1 rpc_name commit rpc_commitid 808464433 rpc_sessid 3745105668
rpc_msgid 6ed74d71-1eda-4757-a4d6-8223b6fca588
```

For other RPCs, the data is reported in the form of XPaths. In this example, the NETCONF operation does not report equivalent CLI command. The RPC name is recorded in the logs.For syslogs with length greater than 400 characters, the log is split into two entries. Here, the XPathis split for brevity

```
RP/0/RP0/CPU0:Mar 15 30 18:39:45.412 UTC: locald_DLRSC[418]: %SECURITY-LOCALD-6-LOCAL_CMD_ACCT
 :
```

```
RPC CMD: rpc_name get by <user> from TTY netconf-921603460 10.0.0.1 rpc_sessid 921603460
rpc_msgid
101 xpath
Cisco-IOS-XR-wdsysmon-fd-proc-oper:process-monitoring/nodes/node[node-name=0/RP0/CPU0]/
process-name/proc-cpu-utilizations/proc-cpu-utilization[process-name=packet]Cisco-IOS-XR-pmengine-oper:
performance management/ethernet/ethernet-ports/ethernet-port/ethernet-current/ethernet-secon


RP/0/RP0/CPU0:Mar 15 18:39:45.412 UTC: locald_DLRSC[418]: %SECURITY-LOCALD-6-LOCAL_CMD_ACCT
 :
RPC CMD: d30/second30-ethersCisco-IOS-XR-pmengine-oper:performance-management/otu/otu-ports/
otu-port/otu-current/otu-minute15/otu-minute15fecsCisco-IOS-XR-wdsysmon-fd-proc-oper:process-monitoring/
nodes/node[node-name=0/RP0/CPU0]/process-name/proc-cpu-utilizations/proc-cpu-utilization[process-name=raw_ip]
```

TACACS+ Logs: The following example shows the logs from a TACACS+ server:

```
Commit changes:

Tue Mar 15 15:56:24 2022 192.0.2.254 root netconf-29961779 192.0.2.1 stop timezone=UTC
task_id=834
service=shell priv-lvl=0 commit_start=2021/10/11 22:56:19.882 commit_id=1000000022 rpc_
sessid=29961779 rpc_msgid=101 rpc_name=commit
Tue Mar 15 15:56:24 2022 192.0.2.254 root netconf-29961779 192.0.2.1 stop timezone=UTC
task_id=835
service=shell priv-lvl=0 cmd=interface GigabitEthernet0/0/0/2 <cr> commit_id=1000000022
rpc_sessid=29961779 rpc_msgid=101 rpc_name=commit
Tue Mar 15 15:56:24 2022 192.0.2.254 root netconf-29961779 192.0.2.1 stop timezone=UTC
task_id=836
service=shell priv-lvl=0 cmd= description test <cr> commit_id=1000000022 rpc_sessid=29961779

rpc_msgid=101 rpc_name=commit
Tue Mar 15 15:56:24 2022 192.0.2.254 root netconf-29961779 192.0.2.1 stop timezone=UTC
task_id=837
service=shell priv-lvl=0 cmd= mtu 1600 <cr> commit_id=1000000022 rpc_sessid=29961779
rpc_msgid=101 rpc_name=commit
Tue Mar 15 15:56:24 2022 192.0.2.254 root netconf-29961779 192.0.2.1 stop timezone=UTC
task_id=838
service=shell priv-lvl=0 cmd= ipv4 address 5.6.7.8 255.255.255.0 route-tag 100 <cr>
commit_id=1000000022
rpc_sessid=29961779 rpc_msgid=101 rpc_name=commit
Tue Mar 15 15:56:24 2022 192.0.2.254 root netconf-29961779 192.0.2.1 stop timezone=UTC
task_id=839
service=shell priv-lvl=0 cmd= shutdown <cr> commit_id=1000000022 rpc_sessid=29961779
rpc_msgid=101 rpc_name=commit
Tue Mar 15 15:56:25 2022 192.0.2.254 root netconf-29961779 192.0.2.1 stop timezone=UTC
task_id=840
service=shell priv-lvl=0 cmd=! <cr> commit_id=1000000022 rpc_sessid=29961779
rpc_msgid=101 rpc_name=commit
Tue Mar 15 15:56:25 2022 192.0.2.254 root netconf-29961779 192.0.2.1 stop timezone=UTC
task_id=841
service=shell priv-lvl=0 commit_end=2021/10/11 22:56:20.471 commit_id=1000000022
rpc_sessid=29961779 rpc_msgid=101 rpc_name=commit

Get-config:

Tue Mar 15 15:05:47 2022 192.0.2.254 root netconf-1616743444 192.0.2.1 stop timezone=UTC
task_id=519
service=shell priv-lvl=0 rpc_sessid=1616743444 rpc_msgid=101 rpc_name=get-config
rpc_xpath= /Cisco-IOS-XR-ifmgr-cfg:interface-configurations
```

Enhancements to Sensor Paths

This section provides an overview about the sensor paths introduced or enhanced across Cisco IOS XR releases.

Table 2: Feature History Table

Feature Name	Release Information	Description
Telemetry Support for OpenConfig Interfaces, IPv4 and IPv6 Addresses and State	Release 7.4.2	This feature provides telemetry GNMI and GRPC support for the following `openconfig-if-ip.yang` sensor paths. Previously, only NETCONF edit-config, get-config and get operations were supported. With this new feature, telemetry polling at a cadence or on-change can be retrieved for IPv4 and IPv6 data. • /oc-if:interfaces/oc-if:interface/oc-if:subinterfaces/oc-if:subinterface/ipv6/ • addresses/address[ip]/state/ip • addresses/address[ip]/state/prefix-length • addresses/address[ip]/state/origin • state/enabled • state/mtu • state/dup-addr-detect-transmits • state/counters/in-pkts • state/counters/in-octets • state/counters/out-pkts • state/counters/out-octets • state/openconfig-if-ip-ext:autoconf/create-global-addresses • /oc-if:interfaces/oc-if:interface/oc-if:subinterfaces/oc-if:subinterface/ipv4/ • addresses/address[ip]/state/ip • addresses/address[ip]/state/prefix-length • addresses/address[ip]/state/origin • state/mtu • state/dhcp-client • state/in-pkts • state/in-octets • state/out-pkts • state/out-octets You can access this data model from the Github repository.

Install Label in oc-platform Data Model

Table 3: Feature History Table

Feature Name	Release Information	Description
Enhancements to openconfig-platform YANG Data Model	Release 7.3.2	The openconfig-platform YANG data model provides a structure for querying hardware and software router components via the NETCONF protocol. This release delivers an enhanced openconfig-platform YANG data model to provide information about: • software version • golden ISO (GISO) label • committed IOS XR packages You can access this data model from the Github repository.

The openconfig-platform (oc-platform.yang) data model is enhanced to provide the following data:

- IOS XR software version (optionally with GISO label)
- Type, description, operational status of the component. For example, a CPU component reports its utilization, temperature or other physical properties.
- List of the committed IOS XR packages

To retrieve oc-platform information from a router via NETCONF, ensure you configured the router with the SH server and management interface:

```
Router#show run
Building configuration...
!! IOS XR Configuration version = 7.3.2
!! Last configuration change at Tue Sep  7 16:18:14 2016 by USER1
!
......
......
netconf-yang agent ssh
ssh server netconf vrf default
interface MgmtEth 0/RP0/CPU0/0
    no shut
    ipv4 address dhcp
```

The following example shows the enhanced `OPERATING_SYSTEM` node component (line card or route processor) of the oc-platform data model:

```
<component>
<name>IOSXR-NODE 0/RP0/CPU0</name>
<config>
<name>0/RP0/CPU0</name>
```

```
</config>
<state>
<name>0/RP0/CPU0</name>
<type xmlns:idx="http://openconfig.net/yang/platform-types">idx:OPERATING_SYSTEM</type>
<location>0/RP0/CPU0</location>
<description>IOS XR Operating System</description>
<software-version>7.3.2</software-version> -----------------------> Label Info
<removable>true</removable>
<oper-status xmlns:idx="http://openconfig.net/yang/platform-types">idx:ACTIVE</oper-status>
</state>
<subcomponents>
 <subcomponent>
  <name><platform>-af-ea-7.3.2v1.0.0.1</name>
  <config>
   <name><platform>-af-ea-7.3.2v1.0.0.1</name>
  </config>
  <state>
   <name><platform>-af-ea-7.3.2v1.0.0.1</name>
  </state>
 </subcomponent>
...
```

The following example shows the enhanced `OPERATING_SYSTEM_UPDATE` package component (RPMs) of the oc-platform data model:

```
<component>
<name>IOSXR-PKG/1 <platform>-isis-2.1.0.0-r732</name>
<config>
<name><platform>-isis-2.1.0.0-r732</name>
</config>
<state>
<name><platform>-isis-2.1.0.0-r732</name>
<type xmlns:idx="http://openconfig.net/yang/platform-types">idx:OPERATING_SYSTEM_UPDATE</type>
<description>IOS XR Operating System Update</description>
<software-version>7.3.2</software-version>-----------------------> Label Info
<removable>true</removable>
<oper-status xmlns:idx="http://openconfig.net/yang/platform-types">idx:ACTIVE</oper-status>
</state>
</component>
```

Associated Commands

- **show install committed**—Shows the committed IOS XR packages.
- **show install committed summary**—Shows a summary of the committed packages along with the committed IOS XR version that is displayed as a label.

OAM for MPLS and SR-MPLS in mpls-ping and mpls-traceroute Data Models

Table 4: Feature History Table

Feature Name	Release Information	Description
YANG Data Models for MPLS OAM RPCs	Release 7.3.2	This feature introduces the `Cisco-IOS-XR-mpls-ping-act` and `Cisco-IOS-XR-mpls-traceroute-act` YANG data models to accommodate operations, administration and maintenance (OAM) RPCs for MPLS and SR-MPLS. You can access these Cisco IOS XR native data models from the Github repository.

The Cisco-IOS-XR-mpls-ping-act and Cisco-IOS-XR-mpls-traceroute-act YANG data models are introduced to provide the following options:

- Ping for MPLS:
    - MPLS IPv4 address
    - MPLS TE
    - FEC-129 Pseudowire
    - FEC-128 Pseudowire
    - Multisegment Pseudowire
- Ping for SR-MPLS:
    - SR policy name or BSID with LSP end-point
    - SR MPLS IPv4 address
    - SR Nil-FEC labels
    - SR Flexible Algorithm
- Traceroute for MPLS:
    - MPLS IPv4 address
    - MPLS TE
- Traceroute for SR-MPLS:
    - SR policy name or BSID with LSP end-point

- SR MPLS IPv4 address
- SR Nil-FEC labels
- SR Flexible Algorithm

The following example shows the ping operation for an SR policy and LSP end-point:

```
<mpls-ping xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-mpls-ping-act">
  <sr-mpls>
  <policy>
    <name>srte_c_10_ep_10.10.10.1</name>
      <lsp-endpoint>10.10.10.4</lsp-endpoint>
  </policy>
  </sr-mpls>
  <request-options-parameters>
    <brief>true</brief>
  </request-options-parameters>
</mpls-ping>
```

Response:

```
<?xml version="1.0"?>
 <mpls-ping-response xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-mpls-ping-act">
  <request-options-parameters>
   <exp>0</exp>
   <fec>false</fec>
   <interval>0</interval>
   <ddmap>false</ddmap>
   <force-explicit-null>false</force-explicit-null>
   <packet-output>
    <interface-name>None</interface-name>
    <next-hop>0.0.0.0</next-hop>
   </packet-output>
   <pad>abcd</pad>
   <repeat>5</repeat>
   <reply>
    <dscp>255</dscp>
    <reply-mode>default</reply-mode>
    <pad-tlv>false</pad-tlv>
   </reply>
   <size>100</size>
   <source>0.0.0.0</source>
   <destination>127.0.0.1</destination>
   <sweep>
    <minimum>100</minimum>
    <maximum>100</maximum>
    <increment>1</increment>
   </sweep>
   <brief>true</brief>
   <timeout>2</timeout>
   <ttl>255</ttl>
  </request-options-parameters>
  <replies>
   <reply>
    <reply-index>1</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
   <reply>
    <reply-index>2</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
   <reply>
    <reply-index>3</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
   <reply>
    <reply-index>4</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
   <reply>
    <reply-index>5</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
  </replies>
 </mpls-ping-response>
```

The following example shows the ping operation for an SR policy BSID and LSP end-point:

```
<mpls-ping xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-mpls-ping-act">
<sr-mpls>
<policy>
    <bsid>1000</bsid>
    <lsp-endpoint>10.10.10.4</lsp-endpoint>
</policy>
</sr-mpls>
<request-options-parameters>
    <brief>true</brief>
</request-options-parameters>
</mpls-ping>
```

Response:

```
<?xml version="1.0"?>
 <mpls-ping-response xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-mpls-ping-act">
  <request-options-parameters>
   <exp>0</exp>
   <fec>false</fec>
   <interval>0</interval>
   <ddmap>false</ddmap>
   <force-explicit-null>false</force-explicit-null>
   <packet-output>
    <interface-name>None</interface-name>
    <next-hop>0.0.0.0</next-hop>
   </packet-output>
   <pad>abcd</pad>
   <repeat>5</repeat>
   <reply>
    <dscp>255</dscp>
    <reply-mode>default</reply-mode>
    <pad-tlv>false</pad-tlv>
   </reply>
   <size>100</size>
   <source>0.0.0.0</source>
   <destination>127.0.0.1</destination>
   <sweep>
    <minimum>100</minimum>
    <maximum>100</maximum>
    <increment>1</increment>
   </sweep>
   <brief>true</brief>
   <timeout>2</timeout>
   <ttl>255</ttl>
  </request-options-parameters>
  <replies>
   <reply>
    <reply-index>1</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
   <reply>
    <reply-index>2</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
   <reply>
    <reply-index>3</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
   <reply>
    <reply-index>4</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
   <reply>
    <reply-index>5</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
  </replies>
 </mpls-ping-response>
```

The following example shows the traceroute operation for an SR policy and LSP end-point:

```
<mpls-traceroute xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-mpls-traceroute-act">
<sr-mpls>
<policy>
     <name>srte_c_10_ep_10.10.10.1</name>
     <lsp-endpoint>10.10.10.4</lsp-endpoint>
</policy>
</sr-mpls>
<request-options-parameters>
    <brief>true</brief>
</request-options-parameters>
```

```
</mpls-traceroute>
```

Response:

```
<?xml version="1.0"?>
 <mpls-traceroute-response xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-mpls-traceroute-act">

   <request-options-parameters>
    <exp>0</exp>
    <fec>false</fec>
    <ddmap>false</ddmap>
    <force-explicit-null>false</force-explicit-null>
    <packet-output>
     <interface-name>None</interface-name>
     <next-hop>0.0.0.0</next-hop>
    </packet-output>
    <reply>
     <dscp>255</dscp>
     <reply-mode>default</reply-mode>
    </reply>
    <source>0.0.0.0</source>
    <destination>127.0.0.1</destination>
    <brief>true</brief>
    <timeout>2</timeout>
    <ttl>30</ttl>
   </request-options-parameters>
   <paths>
    <path>
     <path-index>0</path-index>
     <hops>
     <hop>
         <hop-index>0</hop-index>
         <hop-origin-ip>11.11.11.1</hop-origin-ip>
         <hop-destination-ip>11.11.11.2</hop-destination-ip>
         <mtu>1500</mtu>
         <dsmap-label-stack>
             <dsmap-label>
                 <label>16003</label>
             </dsmap-label>
         </dsmap-label-stack>
         <return-code>0</return-code>
         <return-char> </return-char>
     </hop>
     <hop>
         <hop-index>1</hop-index>
         <hop-origin-ip>11.11.11.2</hop-origin-ip>
         <hop-destination-ip>14.14.14.3</hop-destination-ip>
         <mtu>1500</mtu>
         <dsmap-label-stack>
             <dsmap-label>
                 <label>3</label>
             </dsmap-label>
         </dsmap-label-stack>
         <return-code>8</return-code>
         <return-char>L</return-char>
     </hop>
     <hop>
         <hop-index>2</hop-index>
         <hop-origin-ip>14.14.14.3</hop-origin-ip>
         <hop-destination-ip></hop-destination-ip>
         <mtu>0</mtu>
         <dsmap-label-stack/>
         <return-code>3</return-code>
         <return-char>!</return-char>
```

```
      </hop>
      </hops>
     </path>
    </paths>
    </mpls-traceroute-response>
```

OpenConfig YANG Model:dscp-set

Table 5: Feature History Table

Feature Name	Release Information	Description
OpenConfig YANG Model:dscp-set	Release 7.5.2	This model allows you to configure a minimum and maximum Differentiated Services Code Point (DSCP) value in the dscp-set leaf-list. When you send these values in your request to the NETCONF agent, it filters the traffic by matching the values in the list with the incoming packet header. This ensures that your network is not vulnerable to unwanted traffic. You can access the OC data model from the Github repository.

You can configure two Differentiated Services Code Point (DSCP) values in the dscp-set leaf-list. You can enter these values in any order, and they are internally mapped to dscp-min and dscp-max values. The incoming IPv4 or IPv6 packet header contains the DSCP field. This DSCP field is matched with the range of values that exist between the specified minimum (dscp-min) and maximum (dscp-max) values. When the DSCP field contains one of the values specified in the list, the incoming packet is allowed access to your network. You can add or delete the dscp-set leaf-list in the IPv4 and IPv6 OpenConfig YANG model by sending a NETCONF request.

Note When you delete one of the values from the dscp-set, the model applies the remaining value for both dscp-min and dscp-max fields.

Adding the dscp-set in the IPv4 OC YANG Model

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<edit-config>
  <target>
   <candidate/>
  </target>
  <config type="subtree"xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <acl xmlns="http://openconfig.net/yang/acl">
   <acl-sets>
    <acl-set>
     <name>test-dscp-set</name>
```

```
      <type>ACL_IPV4</type>
      <config>
       <name>test-dscp-set</name>
       <type>ACL_IPV4</type>
      </config>
      <acl-entries>
       <acl-entry>
        <sequence-id>10</sequence-id>
        <config>
         <sequence-id>10</sequence-id>
        </config>
        <actions>
         <config>
          <forwarding-action>ACCEPT</forwarding-action>
         </config>
        </actions>
        <ipv4>
         <config>
          <dscp-set>12</dscp-set>
          <dscp-set>15</dscp-set>
         </config>
        </ipv4>
       </acl-entry>
      </acl-entries>
     </acl-set>
    </acl-sets>
  </acl>
  </config>
</edit-config>
</rpc>
```

Deleting the dscp-set in the IPv4 OC YANG Model

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<edit-config>
<target>
  <candidate/>
</target>
<config type="subtree" xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
 <acl xmlns="http://openconfig.net/yang/acl">
  <acl-sets>
    <acl-set xc:operation="delete">
     <name> test-dscp-set</name>
     <type>ACL_IPV4</type>
   </acl-set>
 </acl-sets>
 </acl>
</config>
</edit-config>
</rpc>
```

Adding the dscp-set in the IPv6 OC YANG Model

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 <edit-config>
<target>
 <candidate/>
</target>
<config type="subtree" xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
 <acl xmlns="http://openconfig.net/yang/acl">
  <acl-sets>
   <acl-set>
    <name>test-dscp-v6-edit</name>
    <type>ACL_IPV6</type>
    <config>
      <name>test-dscp-v6-edit</name>
      <type>ACL_IPV6</type>
   </config>
   <acl-entries>
     <acl-entry>
       <sequence-id>10</sequence-id>
        <config>
           <sequence-id>10</sequence-id>
        </config>
       <actions>
       <config>
         <forwarding-action>ACCEPT</forwarding-action>
      </config>
     </actions>
 <ipv6>
  <config>
    <dscp-set>22</dscp-set>
    <dscp-set>55</dscp-set>
  </config>
 </ipv6>
</acl-entry>
</acl-entries>
 </acl-set>
</acl-sets>
</acl>
</config>
</edit-config>
</rpc>
```

Deleting the dscp-set in the IPv6 OC YANG Model

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<edit-config>
<target>
  <candidate/>
</target>
<config type="subtree" xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
 <acl xmlns="http://openconfig.net/yang/acl">
  <acl-sets>
    <acl-set xc:operation="delete">
     <name>test-dscp-v6-edit</name>
     <type>ACL_IPV6</type>
   </acl-set>
 </acl-sets>
 </acl>
</config>
</edit-config>
</rpc>
```

OpenConfig YANG Model:procmon

Table 6: Feature History Table

Feature Name	Release Information	Description
OpenConfig YANG Model:procmon	Release 7.5.2	This model provides data definitions to monitor the health of one or more processes running on a system, delivering insights into the performance of critical processes and helping remediate performance bottlenecks. For example, the stress tool that is part of the Linux distribution may be consuming high CPU. The openconfig-procmon model pulls this information and sends it to you when you query the node. As a remediation measure, you can then restart the process. You can access the OC data model from the Github repository.

Subscribe to the following sensor path:

openconfig-system:system/processes/process

Based on a Process ID (PID), you can stream state parameters, such as name, args, start-time, uptime, cpu-usage-user, cpu-usage-system, cpu-utilization, memory usage and memory utilization.

When you send the PID to a MDT-capable device requesting state parameters of a process, the PID of the process acts as a key for the request. If the requested PID is invalid, you will not receive any response.

Note The location of the PID is always assumed to be the Active RP. This model does not have any leaf or field where you can specify the location or node name.

Example

This output shows state parameters that monitor the health of the dhcpd process having PID: 22482 using the XR built-in mdt_exec tool. You can also use telemetry tools, such as gNMI and gRPC.

```
RP/0/RP1/CPU0:SF-D#run mdt_exec -s openconfig-system:system/processes/process[pid=22482]
Enter any key to exit...
 Sub_id 200000001, flag 0, len 0
 Sub_id 200000001, flag 4, len 583
--------
{"node_id_str":"SF-D","subscription_id_str":"app_TEST_200000001",
"encoding_path":"openconfig-system:system/processes/process","collection_id":"13",
"collection_start_time":"1648387172382","msg_timestamp":"1648387172384",
"data_json":[{"timestamp":"1648387172384","keys":[{"pid":"22482"}],
"content":{"state":{"pid":"22482","name":"dhcpd","args":["dhcpd"],
"start-time":"1648385883000000000","uptime":"1289384179023","cpu-usage-user":"270000000",
"cpu-usage-system":"180000000","cpu-utilization":0,"memory-usage":"16641952",
"memory-utilization":0}}}],"collection_end_time":"1648387172384"}
--------
 Sub_id 200000001, flag 8, len 0
```