## CONTENTS

### CHAPTER 1

<table>
<thead>
<tr>
<th>Programmability</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>boot ipxe</td>
<td>2</td>
</tr>
<tr>
<td>boot manual</td>
<td>3</td>
</tr>
<tr>
<td>boot system</td>
<td>4</td>
</tr>
<tr>
<td>default boot</td>
<td>5</td>
</tr>
<tr>
<td>dig</td>
<td>7</td>
</tr>
<tr>
<td>guestshell</td>
<td>9</td>
</tr>
<tr>
<td>guestshell portforwarding</td>
<td>10</td>
</tr>
<tr>
<td>install</td>
<td>12</td>
</tr>
<tr>
<td>iox</td>
<td>17</td>
</tr>
<tr>
<td>mlog</td>
<td>18</td>
</tr>
<tr>
<td>netconf legacy</td>
<td>19</td>
</tr>
<tr>
<td>net-dhcp</td>
<td>20</td>
</tr>
<tr>
<td>net-debug</td>
<td>21</td>
</tr>
<tr>
<td>net-show</td>
<td>23</td>
</tr>
<tr>
<td>net-tcp-bufs</td>
<td>24</td>
</tr>
<tr>
<td>net-tcp-mss</td>
<td>25</td>
</tr>
<tr>
<td>net6-dhcp</td>
<td>26</td>
</tr>
<tr>
<td>net6-show</td>
<td>27</td>
</tr>
<tr>
<td>ping</td>
<td>28</td>
</tr>
<tr>
<td>ping4</td>
<td>29</td>
</tr>
<tr>
<td>ping6</td>
<td>30</td>
</tr>
<tr>
<td>show install</td>
<td>31</td>
</tr>
<tr>
<td>show platform yang-management process</td>
<td>34</td>
</tr>
<tr>
<td>show telemetry ietf subscription</td>
<td>36</td>
</tr>
</tbody>
</table>
Programmability

- boot ipxe, on page 2
- boot manual, on page 3
- boot system, on page 4
- default boot, on page 5
- dig, on page 7
- guestshell, on page 9
- guestshell portforwarding, on page 10
- install, on page 12
- iox, on page 17
- mlog, on page 18
- netconf legacy, on page 19
- net-dhcp, on page 20
- net-debug, on page 21
- net-show, on page 23
- net-tcp-bufs, on page 24
- net-tcp-mss, on page 25
- net6-dhcp, on page 26
- net6-show, on page 27
- ping, on page 28
- ping4, on page 29
- ping6, on page 30
- show install, on page 31
- show platform yang-management process, on page 34
- show telemetry ietf subscription, on page 36
**boot ipxe**

To configure iPXE boot, use the **boot ipxe** command in global configuration mode. To disable the configuration, use the **no** form of this command.

```
boot ipxe {forever |timeout seconds} switch switch-number
no boot ipxe {forever |timeout seconds} switch switch-number
```

**Syntax Description**

- **forever**
  - Attempts iPXE boot forever.
- **timeout seconds**
  - Configures a timeout in seconds for iPXE network boot. Valid values are from 1 to 2147483647.
- **switch switch-number**
  - Enables iPXE boot for switches in the stack. Valid values are from 0 to 9.

**Command Modes**

Global configuration (config)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Denali 16.3.2</td>
<td>This command was introduced on Cisco Catalyst 3650 and 3850 Series Switches.</td>
</tr>
<tr>
<td>Cisco IOS XE Everest 16.6.1</td>
<td>This command was implemented on Cisco Catalyst 9300 and 9500 Series Switches</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

iPXE is an open source implementation of the Preboot eXecution Environment (PXE). Bootloaders boot an image located on an HTTP, FTP, or a TFTP server.

If the **forever** keyword is configured, the switch sends Dynamic Host Configuration Protocol (DHCP) requests forever. If the **timeout** keyword is configured, DHCP requests are sent for the specified amount of time, and when the timeout expires, the switch reverts to device boot.

**Example**

The following example shows how to configure an iPXE boot timeout for switch 2:

```
Device(config)# boot ipxe timeout 240 switch 2
```
boot manual

To configure manual boot, use the `boot manual` command in global configuration mode. To remove the configuration, use the `no` form of this command.

```
boot manual switch switch-number
no boot manual switch switch-number
```

**Syntax Description**

- `switch switch-number` Configures manual boot for the switches in the stack.

**Command Default**

Manual boot is enabled.

**Command Modes**

Global configuration (config)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Denali 16.3.2</td>
<td>This command was introduced on Cisco Catalyst 3650 and 3850 Series Switches.</td>
</tr>
<tr>
<td>Cisco IOS XE Everest 16.6.1</td>
<td>This command was implemented on Cisco Catalyst 9300 and 9500 Series Switches</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

When manual boot is disabled, and the switch reloads, the boot process starts automatically. When manual boot is disabled, the bootloader determines whether to execute a device boot or a network boot based on the configured value of the iPXE ROMMON variable.

**Example**

The following example shows how to configure manual boot for switch 2:

```
Device(config)# boot manual switch 2
```
To enable a system image boot, use the `boot system` command in global configuration mode. To disable the configuration, use the `no` form of this command.

```
boot system switch {all number} {flash: |ftp: |http: |tftp:}
```

```
no boot system [{switch |{all number}}] [{flash: |ftp: |http: |tftp:}]
```

### Syntax Description

- **flash:** Specifies the flash filesystem to boot an image.
- **ftp:** Specifies an FTP location to boot an image.
- **http:** Specifies an HTTP location to boot an image.
- **tftp:** Specifies a TFTP location to boot an image.
- **switch number** Enables booting for switches in a stack. Valid values are from 0 to 9.

### Command Modes

Global configuration (config)

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Denali 16.3.2</td>
<td>This command was introduced on Cisco Catalyst 3650 and 3850 Series Switches.</td>
</tr>
<tr>
<td>Cisco IOS XE Everest 16.6.1</td>
<td>This command was implemented on Cisco Catalyst 9300 and 9500 Series Switches</td>
</tr>
</tbody>
</table>

### Usage Guidelines

You can either use an IPv4 or an IPv6 address for the remote FTP/HTTP/TFTP servers. When using an IPv6 address, you must enter the IPv6 address inside square brackets (as per RFC 2732); otherwise, the device will not boot.

**Note**

IPv6 is not supported on Catalyst 9000 Series Switches.

### Example

The following example shows how to boot an image from an IPv4 HTTP server:

```
Device(config)# boot system switch 1 http://192.0.2.42/image-filename
```

The following example shows how to boot an image from an IPv6 HTTP server:

```
Device(config)# boot system switch 1 http://[2001:db8::1]/image-filename
```
default boot

To modify the default boot system parameters, use the `default boot` command in global configuration mode.

```
default boot {ipxe {forever |timeout |seconds} |manual |system {flash: |ftp: |http: |tftp:}} switch number
```

**Syntax Description**

- **ipxe**: Enables iPXE boot.
- **forever**: Attempts iPXE boot forever.
- **timeout seconds**: Configures a boot timeout in seconds. Valid values are from 1 to 2147483647.
- **manual**: Enables manual boot.
- **system**: Enables a system image boot.
- **flash:** Specifies the flash filesystem to boot an image.
- **ftp:** Specifies an FTP location to boot an image.
- **http:** Specifies an HTTP location to boot an image.
- **tftp:** Specifies a TFTP location to boot an image.
- **switch number**: Enables booting for switches in a stack. Valid values are from 0 to 9.

**Command Default**

Device boot is enabled.

**Command Modes**

Global configuration (config)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Denali 16.3.2</td>
<td>This command was introduced on Cisco Catalyst 3650 and 3850 Series Switches.</td>
</tr>
<tr>
<td>Cisco IOS XE Everest 16.6.1</td>
<td>This command was implemented on Cisco Catalyst 9300 and 9500 Series Switches</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

You can either use the `no boot ipxe` or the `default boot ipxe` command to configure device boot.

If the `forever` keyword is configured, the switch sends Dynamic Host Configuration Protocol (DHCP) requests forever. If the `timeout` keyword is configured, DHCP requests are sent for the specified amount of time, and when the timeout expires, the switch reverts to device boot.

**Example**

The following example shows how to enable the default boot mode:

```
```
Device(config)# default boot ipxe
dig

To do a lookup of the Domain Name System (DNS) server, use the `dig` command in rommon mode.

```
dig hostname {v4 v6} [{dns-server-address}]
```

### Syntax Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname</td>
<td>DNS host name</td>
</tr>
<tr>
<td>v4</td>
<td>IPv4 address.</td>
</tr>
<tr>
<td>v6</td>
<td>IPv6 address.</td>
</tr>
<tr>
<td>dns-server-address</td>
<td>(Optional) DNS Server IP address.</td>
</tr>
</tbody>
</table>

### Command Modes

Rommon

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

This command does a look up of the DNS name and displays the IP/IPv6 address of the DNS server.

### Example

The following is sample output from the `dig hostname` command:

```
Device: dig example.org
DNS lookup using 2001:DB8::1
addr = 2001:DB8:0000:0000:0000:0000:0000:0001
```

The following is sample output from the `dig hostname v4` command:

```
Device: dig example.org v4
DNS lookup using 10.29.27.5
addr = 172.16.0.1
```

The following is sample output from the `dig hostname v4 dns-server-address` command:

```
Device: dig example.org v4 10.29.27.5
DNS lookup using 10.29.27.5
addr = 172.16.0.1
```

The following is sample output from the `dig hostname v6` command:

```
Device: dig example.org v6
DNS lookup using 2001:DB::1
addr = 2001:DB8:0000:0000:0000:0000:0000:0001
```
<table>
<thead>
<tr>
<th>Related Commands</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>net-debug</td>
<td>Displays or changes the network debug values.</td>
</tr>
</tbody>
</table>
guestshell

To configure Guestshell infrastructure functionality, use the `guestshell` command in Privileged EXEC mode.

```
guestshell {destroy |disable |enable |run [{linux-executable}]} 
```

**Syntax Description**
- `destroy` Deactivates and uninstalls the Guestshell service.
- `disable` Disables the Guestshell service.
- `enable` Disables the Guestshell service.
- `run` [`linux-executable`] Executes or runs a Linux program in the Guestshell.

**Command Default**
Privileged EXEC (#)

**Command Modes**
- Privileged EXEC (#)

**Command History**
- Cisco IOS XE Everest 16.5.1
  - This command was introduced.

**Usage Guidelines**
Guestshell is an embedded Linux environment that allows customers to develop and run custom Python applications for automated control and management of Cisco switches. Guestshell is packaged as a Cisco application hosting framework (CAF)-formatted tar file (guest_shell.tar) into the Cisco IOS XE Everest 16.5.x release image read-only file system.

Configure the `iox` command in global configuration mode, before configuring this command. IOx is the Cisco-developed framework for hosting customer-deployed Linux applications on Cisco networking systems.

**Example**
The following example shows how to enable and run the Guestshell:

```
Device# configure terminal
Device(config)# iox
Device(config)# exit
Device# guestshell enable
Device# guestshell run
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>iox</td>
<td>Configure IOx services.</td>
</tr>
</tbody>
</table>
guestshell portforwarding

To enable Guest Shell port forwarding, use the **guestshell portforwarding** command in privileged EXEC mode.

```
guestshell portforwarding {add table-entry entry-name service {tcp | udp} source-port port-number destination-port port-number | delete table-entry entry-name }
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>add</strong></td>
<td>Adds an IP table entry.</td>
</tr>
<tr>
<td><strong>table-entry entry-name</strong></td>
<td>Specifies the IP table name. The <strong>table-name</strong> argument must be unique, and it can be alphanumeric characters.</td>
</tr>
<tr>
<td><strong>service</strong></td>
<td>Specifies the service protocol.</td>
</tr>
<tr>
<td><strong>tcp</strong></td>
<td>Specifies TCP as the service protocol.</td>
</tr>
<tr>
<td><strong>udp</strong></td>
<td>Specifies UDP as the service protocol.</td>
</tr>
<tr>
<td><strong>source-port port-number</strong></td>
<td>Specifies the source port. Valid values for the <strong>port-number</strong> argument are from 1 to 65535.</td>
</tr>
<tr>
<td><strong>destination-port port-number</strong></td>
<td>Specifies the destination port. Valid values for the <strong>port-number</strong> argument are from 1 to 65535.</td>
</tr>
<tr>
<td><strong>delete</strong></td>
<td>Deletes an IP table entry.</td>
</tr>
</tbody>
</table>

**Command Default**

Port forwarding is not enabled.

**Command Modes**

Privileged EXEC (#)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.6.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use this command to enable port forwarding for Guest Shell, when it connected through the GigabitEthernet 0/0 management interface.

**Examples**

The following example shows how to enable port forwarding for Guest Shell:

```
Device# configure terminal
Device(config)# iox
```
Device(config)# exit
Device# guestshell portforwarding add table-entry table1 service tcp
   source-port 32 destination-port 9
Device#

The following example shows how to disable port forwarding for Guest Shell:

Device# guestshell portforwarding delete table-entry table1
Device#

<table>
<thead>
<tr>
<th>Related Commands</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>guestshell</td>
<td>Configures the Guest Shell infrastructure functionality.</td>
</tr>
</tbody>
</table>
To install data model update packages, use the `install` command in privileged EXEC mode.

```plaintext

Syntax Description

| activate | Validates whether the model update package is added through the install add command, and restarts the NETCONF processes (confd and opdatamgrd). This keyword runs a compatibility check, updates package status, and if the package can be restarted, it triggers post-install scripts to restart the necessary processes, or triggers a reload for non-restartable packages. |
| file | Specifies the package to be activated. |
| prompt-level {all | none} | (Optional) Prompts the user about installation activities. For example, the activate keyword, automatically triggers a reload for packages that require a reload. Before activating the package, a message will prompt users as to whether they want to continue. The all keyword allows you to enable prompts. The none keyword disables prompts. |
| add | Copies files from a remote location (via FTP, TFTP) to a device, and performs a compatibility check for the platform and image versions. This keyword runs base compatibility checks to ensure that a specified package is supported on a platform. It also adds an entry in the package file, so that the status can be monitored and maintained. |
| {http: | https: | rcp: | scp: | tftp:} | Specifies the package to be added. |
### commit
Makes changes persistent over reloads.
You can do a commit after activating a package, while the system is up, or after the first reload. If a package is activated, but not committed, it remains active after the first reload, but not after the second reload.

### deactivate
Deactivates an installed package.
Deactivating a package also updates the package status and triggers a process restart or a reload.

### remove
Remove installed packages.
The package file is removed from the file system. The `remove` keyword can only be used on packages that are currently inactive.

### inactive
Removes all inactive packages from the device.

### rollback
Rollbacks the data model update package to the base version, the last committed version, or a known commit ID, and restarts NECONF processes.

### to base
Returns to the base image.

### committed
Returns to the installation state when the last commit operation was performed.

### id install-ID
Returns to the specific install point ID. Valid values are from 1 to 4294967295.

---

**Command Default**
Model update packages are not installed.

**Command Modes**
Privileged EXEC (#)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1</td>
<td>This command was introduced on the following platforms:</td>
</tr>
<tr>
<td></td>
<td>• Cisco 4000 Series Integrated Services Routers</td>
</tr>
<tr>
<td></td>
<td>• Cisco Catalyst 9300 Series Switches</td>
</tr>
<tr>
<td></td>
<td>• Cisco Catalyst 9500 Series Switches</td>
</tr>
<tr>
<td></td>
<td>• Cisco Cloud Services Router 1000v</td>
</tr>
<tr>
<td></td>
<td>• Cisco Integrated Services Virtual Routers (ISRv)</td>
</tr>
</tbody>
</table>

In Service Model Update adds new data models or extend functionality to existing data models. The update package provides YANG model enhancements outside of a release cycle. The update package is a superset of all existing models; it includes all existing models as well as updated YANG models.

A model update package must be added prior to activating the update package. A package must be deactivated, before it is removed from the bootflash.

**Cisco 4000 Series Integrated Services Routers**

The following example shows how to add an install package on a device:

```
Device# install add file tftp://172.16.0.1/tftpboot/folder1/isr4300-universalk9.2017-01-10_13.15.1.CSCxxxxxxx.dmp.bin
```

```
install add: START Sun Feb 26 05:57:04 UTC 2017
Downloading file tftp://172.16.0.1/tftpboot/folder1/isr4300-universalk9.2017-01-10_13.15.1.CSCxxxxxxx.dmp.bin
Finished downloading file tftp://172.16.0.1/tftpboot/folder1/isr4300-universalk9.2017-01-10_13.15.1.CSCxxxxxxx.dmp.bin
SUCCESS: install_add /bootflash/isr4300-universalk9.2017-01-10_13.15.1.CSCxxxxxxx.dmp.bin
```

```
Sun Feb 26 05:57:22 UTC 2017
```

The following example shows how to activate an install package:

```
Device# install activate file bootflash:
isr4300-universalk9.2017-01-10_13.15.1.CSCxxxxxxx.dmp.bin
```

```
install_activate: START Sun Feb 26 05:58:41 UTC 2017
DMP package.
Netconf processes stopped
SUCCESS: install_activate /bootflash/isr4300-universalk9.2017-01-10_13.15.1.CSCxxxxxxx.dmp.bin
```

```
Sun Feb 26 05:58:58 UTC 2017
```

The following example shows how to commit an installed package:

```
Programmability Command Reference, Cisco IOS XE Fuji 16.7.1
```

```
*Feb 26 05:58:47.661: %DMI-4-SUB_READ_FAIL: SIP0: vtyserverutild:
Confd subscription socket read failed Lost connection to ConfD (45): EOF on socket to ConfD.
*Feb 26 05:59:43.269: %DMI-5-SYNC_COMPLETE: SIP0: syncfd:
The running configuration has been synchronized to the NETCONF running data store.
```

The following example shows how to commit an installed package:
**Device**

```
install commit
```

*install_commit: START Sun Feb 26 06:46:48 UTC 2017
SUCCESS: install_commit Sun Feb 26 06:46:52 UTC 2017

The following example shows how to rollback to the base package:

**Device**

```
install rollback to base
```

*install_rollback: START Sun Feb 26 06:50:29 UTC 2017
7 install_rollback: Restarting impacted processes to take effect
7 install_rollback: restarting confd

*Feb 26 06:50:34.957: %DMI-4-CONTROL_SOCKET_CLOSED: SIP0: syncfd:
  Confd control socket closed Lost connection to ConfD (45): EOF on socket to ConfD.
*Feb 26 06:50:34.962: %DMI-4-CONTROL_SOCKET_CLOSED: SIP0: nesd:
  Confd control socket closed Lost connection to ConfD (45): EOF on socket to ConfD.
*Feb 26 06:50:34.963: %DMI-4-SUB_READ_FAIL: SIP0: vtyserverutil:
  Confd subscription socket read failed Lost connection to ConfD (45):
  EOF on socket to ConfD.Netconf processes stopped
7 install_rollback: DMP activate complete
SUCCESS: install_rollback Sun Feb 26 06:50:41 UTC 2017

*Feb 26 06:51:28.901: %DMI-5-SYNC_START: SIP0: syncfd:
  External change to running configuration detected.
The running configuration will be synchronized to the NETCONF running data store.
*Feb 26 06:51:30.339: %DMI-5-SYNC_COMPLETE: SIP0: syncfd:
  The running configuration has been synchronized to the NETCONF running data store.

**Cisco Catalyst 3000 Series Switches**

The following example shows how to add an install package on a device:

**Device**

```
install add file tftp://172.16.0.1//tftpboot/folder1/cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
```

*install_add: START Sat Jul 29 05:57:04 UTC 2017
Download file tftp://172.16.0.1//tftpboot/folder1/cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
Finished downloading file tftp://172.16.0.1//tftpboot/folder1/cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin to bootflash:cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
SUCCESS: install_add /bootflash/... Sat Jul 29 05:57:22 UTC 2017

The following sample output from the **show install summary** command displays that the update package is now committed, and that it will be persistent across reloads:

**Device**

```
show install summary
```

Active Packages:
bootflash:cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
Inactive Packages:
No packages
Committed Packages:
bootflash:cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
Uncommitted Packages:
No packages

Device
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show install</td>
<td>Displays information about model update packages.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show install</td>
<td>Displays information about data model update packages.</td>
</tr>
</tbody>
</table>
**iox**

To configure IOx services, use the `iox` command in global configuration mode. To remove the configuration, use the `no` form of this command.

```plaintext
iox
no iox
```

This command has no arguments or keywords.

<table>
<thead>
<tr>
<th>Command Default</th>
<th>Command Modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global configuration (config)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command History</th>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOx</td>
<td>Cisco IOS XE Everest 16.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

IOx is the Cisco-developed framework for hosting customer-deployed Linux applications on Cisco networking systems. The `iox` command must be configured before configuring the `guestshell` command.

The following example shows how to configure IOx services:

```plaintext
Device# configure terminal
Device(config)# iox
Device(config)# exit
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>guestshell</td>
<td>Configures Guestshell infrastructure functionality.</td>
</tr>
</tbody>
</table>
To direct log messages to a memory buffer instead of the serial port, use the `mlog` command in rommon mode.

```
mlog  [{show | reset | ctrl [{on | off | toggle}]}]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show</code></td>
<td>(Optional) Displays memory log messages.</td>
</tr>
<tr>
<td><code>reset</code></td>
<td>(Optional) Resets the logging of messages to the memory log.</td>
</tr>
<tr>
<td><code>ctrl</code></td>
<td>(Optional)</td>
</tr>
<tr>
<td><code>on</code></td>
<td>(Optional)</td>
</tr>
<tr>
<td><code>off</code></td>
<td>(Optional)</td>
</tr>
<tr>
<td><code>toggle</code></td>
<td>(Optional)</td>
</tr>
</tbody>
</table>

**Command Modes**

Rommon

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

This command directs protocol log (that is all logs controlled by the `net-debug` command) messages to a memory buffer instead of the serial port.

With memory logging, log messages are displayed after a test is run. For example, HTTP debugs can be enabled through memory logging. Log messages are displayed in the memory buffer after running a copy from `http://server/name to null:` command.

**Example**

The following example shows how to direct log messages to the memory buffer:

```
Device:  mlog show
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>net-debug</code></td>
<td>Displays or changes the network debug values.</td>
</tr>
</tbody>
</table>
**netconf legacy**

To enable legacy NETCONF protocol, use the `netconf legacy` command in global configuration mode. To disable the legacy NETCONF protocol, use the `no` form of this command.

```
netconf legacy
no netconf legacy
```

This command has no arguments or keywords.

**Command Default**

Legacy NETCONF protocol is not enabled.

**Command Modes**

Global configuration (config)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Denali 16.3.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

If this command is enabled, the RFC-compliant NETCONF client (ncclient) does not work. This command enables the legacy NETCONF protocol that is non-RFC-compliant.

**Example**

The following example shows how to disable the legacy NETCONF protocol:

```
Device> enable
Device# configure terminal
Device(config)# no netconf legacy
```
To initiate an IPv4 Dynamic Host Control Protocol (DHCP) request for remote configuration, use the `net-dhcp` command in rommon mode.

```
net-dhcp [timeout]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout</td>
<td>(Optional) Timeout in seconds.</td>
</tr>
</tbody>
</table>

**Command Modes**

Rommon

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

This command initiates an IPv4 DHCP request and processes the reply.

**Example**

The following example shows how to enable the `net-dhcp` command:

```
Device: net-dhcp
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>net-debug</td>
<td>Displays or changes the network debug values.</td>
</tr>
<tr>
<td>net-show</td>
<td>Displays network parameters.</td>
</tr>
<tr>
<td>net6-dhcp</td>
<td>Initiates an IPv6 DHCP request for remote configuration.</td>
</tr>
</tbody>
</table>
net-debug

To display or change the network debug values use the **net-debug** command in rommon mode.

**net-debug** [{new-value}]

**Syntax Description**

| new-value | (Optional) New debug value to use. |

**Command Modes**

Rommon

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

This command enables or disables log levels for each of the following functional areas:

- Domain Name System (DNS)
- Dynamic Host Control Protocol (DHCP)
- File Transfer Protocol (FTP)
- Hypertext Transfer Protocol (HTTP)
- IP
- TCP
- UDP
- Uniform Resource Identifier (URI)

**Example**

This following is sample output from the **net-debug** command:

```
Device: net-debug

ether: 0
  ip: 0
dhcp: 0
udp: 0
tcp: 0
http: 0
dns: 0
uri: 0
t/ftp: 2
ip6: 0
dhcp6: 0:000 200 000 000
```
### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mlog</td>
<td>Directs log messages to a memory buffer instead of the serial port.</td>
</tr>
</tbody>
</table>
net-show

To display network parameters, use the **net-show** command in rommon mode.

**net-show**

This command has no arguments or keywords.

## Command Modes

<table>
<thead>
<tr>
<th>Command Modes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Rommon</td>
<td></td>
</tr>
</tbody>
</table>

## Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

## Usage Guidelines

This command displays network configuration such as IP address, gateway, MAC address and so on.

### Example

The following is sample output from the **net-show** command:

```
Device:  net-show
Network params:
IPv4:     ip addr 10.29.27.150
           netmask 255.255.0.0
           gateway 10.29.0.1
IPv6:
       link-local addr fe80::366f:90ff:feb8:cb80
       site-local addr fec0::366f:90ff:feb8:cb80
       DHCP addr 2001:dead:beef:cafe::9999
       router addr fe80::7ada:6eff:fe13:8580
       SLAAC addr 2001:dead:beef:cafe:90ff:feb8:cb80 /64
       SLAAC addr f00d::366f:90ff:feb8:cb80 /64
       SLAAC addr feed::366f:90ff:feb8:cb80 /64
Common:  macaddr 34:6f:90:b8:cb:80
       dns 2001:dead:beef:cafe::5
       bootfile http://www.example.org/ed10m
       domain ip6.example.org
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>net6-show</td>
<td>Displays IPv6 network parameters.</td>
</tr>
</tbody>
</table>
To display TCP buffers, use the `net-tcp-bufs` command in rommon mode.

```
net-tcp-bufs [{mss}]
```

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>mss</code></td>
<td>(Optional) The Maximum Segment Size (MSS) of TCP buffers.</td>
</tr>
</tbody>
</table>

### Command Modes

- Rommon

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

You can set the MSS of TCP buffers using the `mss` argument.

### Example

The following is sample output from the `net-tcp-bufs` command:

```
Device: net tcp-bufs
tcp_num_buffs 4
```

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>net-tcp-mss</td>
<td>View or set the TCP MSS.</td>
</tr>
</tbody>
</table>
net-tcp-mss

To view or set the TCP Maximum Segment Size (MSS), use the net-tcp-mss command in rommon mode.

```
net-tcp-mss [mss]
```

**Syntax Description**

| mss | (Optional) The Maximum Segment Size (MSS) of TCP buffers. |

**Command Modes**

Rommon

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `mss` argument to change the MSS size.

**Example**

The following is sample output from the net-tcp-mss command:

```
Device: net-tcp-mss

switch: net-tcp-mss
tcp_segment_size 1024
```

The following is sample output from the net-tcp-mss mss command:

```
Device: net-tcp-mss 700

switch: net-tcp-mss 700
tcp_segment_size 700
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>net-tcp-bufs</td>
<td>Displays TCP buffers.</td>
</tr>
</tbody>
</table>
**net6-dhcp**

To initiate an IPv6 Dynamic Host Control Protocol (DHCP) request for remote configuration, use the `net6-dhcp` command in rommon mode.

```
net6-dhcp  [{timeout}]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout</td>
<td>(Optional) Timeout in seconds.</td>
</tr>
</tbody>
</table>

**Command Modes**

<table>
<thead>
<tr>
<th>Command Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rommon</td>
<td></td>
</tr>
</tbody>
</table>

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

You can change the timeout by specifying a time in seconds.

**Example**

The following example shows how to enable the `net6-dhcp` command:

Device: `net6-dhcp`

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>net-debug</td>
<td>Displays or changes the network debug values.</td>
</tr>
<tr>
<td>net-dhcp</td>
<td>Initiates an IPv4 DHCP request and processes the reply.</td>
</tr>
<tr>
<td>net-show</td>
<td>Displays network parameters.</td>
</tr>
</tbody>
</table>
net6-show

To display IPv6 network parameters, use the **net6-show** command in rommon mode.

**net6-show**

This command has no arguments or keywords.

**Command Modes**

Rommon

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>16.5.1</td>
<td></td>
</tr>
</tbody>
</table>

**Usage Guidelines**

**Example**

The following is sample output from the **net6-show** command:

```
Device: net6-show
switch: net6-show
IP6 addresses
link-local addr fe80::366f:90ff:feb8:cb80
site-local addr fec0::366f:90ff:feb8:cb80
    DHCP addr 2001:dead:beef:cafe::9999
    DHCP addr 2001:dead:beef:cafe::9999
    SLAAC addr 2001:dead:beef:cafe::9999 /64
    SLAAC addr 2001:dead:beef:cafe::9999 /64
    SLAAC addr 2001:dead:beef:cafe::9999 /64
    null addr ::
    all-nodes addr ff02::1
    all-routers addr ff02::2
    all-dhcp addr ff02::1
    slct-node addr ff02::1:ff8:cb80
    ll mmac addr 33:33:00:00:00:01
    sl mmac addr 33:33:00:00:00:02
    sn mmac addr 33:33:ff:b8:cb:80
    dhcp mmac addr 33:33:ff:00:99:99

IP6 neighbour table
0: ip6 fec0::366f:90ff:feb8:cb80 MAC 34:6f:90:b8:cb:80
4: ip6 fe80::32f7:9ff:fe08:7ebd MAC 30:f7:0d:08:7e:bd
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>net-show</td>
<td>Displays network parameters.</td>
</tr>
</tbody>
</table>
The ping command is a very common method for troubleshooting the accessibility of devices. A timeout is implemented at the bootloader device prompt, that allows the bootloader to poll the TCP stack every 200 ms. As a result, the bootloader may take up to 200 ms to respond to pings. However, when the bootloader is downloading a file, and thus actively polling for new packets, it responds to ping quickly.

Example

The following is sample output from the ping command:

Device: ping 10.29.27.5

Ping 10.29.27.5 with 32 bytes of data ...
Host 10.29.27.5 is alive.

The following is sample output from the ping host_ip_address retries command:

Device: ping 10 6.29.27.5 6

Ping 10.29.27.5 with 32 bytes of data ... reply received in 0 ms
Ping 10.29.27.5 with 32 bytes of data ... reply received in 0 ms
Ping 10.29.27.5 with 32 bytes of data ... reply received in 0 ms
Ping 10.29.27.5 with 32 bytes of data ... reply received in 0 ms
Ping 10.29.27.5 with 32 bytes of data ... reply received in 0 ms
Ping 10.29.27.5 with 32 bytes of data ... reply received in 0 ms

Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ping</td>
<td>Diagnoses basic network connectivity.</td>
</tr>
<tr>
<td>ping4</td>
<td>Determines the network connectivity to another device using IPv6 addressing.</td>
</tr>
</tbody>
</table>
ping4

To diagnose basic network connectivity, use the `ping4` command in rommon mode.

```
ping4 [{host_ip_address}][{retries}]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>host_ip_address</code></td>
<td>(Optional) IP address of the host to be pinged.</td>
</tr>
<tr>
<td><code>retries</code></td>
<td>(Optional) Number of retries.</td>
</tr>
</tbody>
</table>

**Command Modes**

Rommon

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `ping` and `ping4` commands are the same

A timeout is implemented at the bootloader device prompt, that allows the bootloader to poll the TCP stack every 200 ms. As a result, the bootloader may take up to 200 ms to respond to pings. However, when the bootloader is downloading a file, and thus actively polling for new packets, it responds to ping quickly.

**Example**

The following is sample output from the `ping4 host_ip_address` command:

```
Device: ping4 10.29.27.5

Ping 10.29.27.5 with 32 bytes of data ...
Host 10.29.27.5 is alive.
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ping</td>
<td>Diagnoses basic network connectivity.</td>
</tr>
<tr>
<td>ping6</td>
<td>Determines the network connectivity to another device using IPv6 addressing.</td>
</tr>
</tbody>
</table>
ping6

To determine the network connectivity to another device using IPv6 addressing, use the ping6 command, rommon mode.

```plaintext
ping6  [{host}]  [{repeats}]  [{len}]
```

**Syntax Description**

- **host** (Optional) IP address of the host to be pinged.
- **repeats** (Optional) Number of times to repeat the ping.
- **len**

**Command Modes**

Rommon

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

A timeout is implemented at the bootloader device prompt, that allows the bootloader to poll the TCP stack every 200 ms. As a result, the bootloader may take up to 200 ms to respond to pings. However, when the bootloader is downloading a file, and thus actively polling for new packets, it responds to ping quickly.

**Example**

The following is sample output from the ping6 **host retries len** command:

```
Device: ping6 2001:dead:beef:cafe::5 6 1000
Ping host 2001:dead:beef:cafe::5, 6 times, 1000 bytes
Pinging 2001:dead:beef:cafe::5 ... reply in 0 ms
Pinging 2001:dead:beef:cafe::5 ... reply in 1 ms
Pinging 2001:dead:beef:cafe::5 ... reply in 1 ms
Pinging 2001:dead:beef:cafe::5 ... reply in 0 ms
Pinging 2001:dead:beef:cafe::5 ... reply in 0 ms
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ping</td>
<td>Diagnoses basic network connectivity.</td>
</tr>
<tr>
<td>ping4</td>
<td>Diagnoses basic network connectivity.</td>
</tr>
</tbody>
</table>
show install

To display information about data model update packages, use the `show install` command in privileged EXEC mode.

```
show install {active |committed |inactive |log |package {bootflash: |flash: |webui:}|rollback |summary |uncommitted}
```

### Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>Displays information about active packages.</td>
</tr>
<tr>
<td>committed</td>
<td>Displays package activations that are persistent.</td>
</tr>
<tr>
<td>inactive</td>
<td>Displays inactive packages.</td>
</tr>
<tr>
<td>log</td>
<td>Displays entries stored in the logging installation buffer.</td>
</tr>
<tr>
<td>package</td>
<td>Displays metadata information about the package, including description, restart information, components in the package, and so on.</td>
</tr>
<tr>
<td>`{bootflash:</td>
<td>flash:</td>
</tr>
<tr>
<td>rollback</td>
<td>Displays the software set associated with a saved installation.</td>
</tr>
<tr>
<td>summary</td>
<td>Displays information about the list of active, inactive, committed, and superseded packages.</td>
</tr>
<tr>
<td>uncommitted</td>
<td>Displays package activations that are non persistent.</td>
</tr>
</tbody>
</table>

### Command Modes

Privileged EXEC (#)

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1</td>
<td>This command was introduced on the following platforms:</td>
</tr>
<tr>
<td></td>
<td>• Cisco 4000 Series Integrated Services Routers</td>
</tr>
<tr>
<td></td>
<td>• Cisco Catalyst 9300 Series Switches</td>
</tr>
<tr>
<td></td>
<td>• Cisco Catalyst 9500 Series Switches</td>
</tr>
<tr>
<td></td>
<td>• Cisco Cloud Services Router 1000v</td>
</tr>
<tr>
<td></td>
<td>• Cisco Integrated Services Virtual Routers (ISRv)</td>
</tr>
<tr>
<td>Cisco IOS XE Everest 16.6.1</td>
<td>This command was implemented on the following platforms:</td>
</tr>
<tr>
<td></td>
<td>• Cisco Catalyst 3650 Series Switches</td>
</tr>
<tr>
<td></td>
<td>• Cisco Catalyst 3850 Series Switches</td>
</tr>
</tbody>
</table>
Usage Guidelines

Use the show commands to view the status of an installed model update package.

**Cisco 4000 Series Integrated Services Routers**

The following is sample output from the `show install package` command:

```
Device# show install package bootflash:
isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
```

Name: isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Version: 16.5.1.0.199.1484082952..Everest
Platform: ISR4300
Package Type: dmp
Defect ID: CSCxxxxxxx
Package State: Added
Supersedes List: {}
Smu ID: 1
Device#

The following is sample output from the `show install summary` command:

```
Device# show install summary
Active Packages: bootflash:isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Inactive Packages: No packages
Committed Packages: No packages
Uncommitted Packages: bootflash:isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Device#
```

The following is sample output from the `show install log` command:

```
Device# show install log
[0|install_op_boot]: START Fri Feb 24 19:20:19 Universal 2017
[0|install_op_boot]: END SUCCESS Fri Feb 24 19:20:23 Universal 2017
[3|install_add]: START Sun Feb 26 05:55:31 UTC 2017
[3|install_add]: END SUCCESS Sun Feb 26 05:57:22 UTC 2017
[4|install_add]: END SUCCESS /bootflash/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Device# show install log
[0|install_op_boot]: START Fri Feb 24 19:20:19 Universal 2017
[0|install_op_boot]: END SUCCESS Fri Feb 24 19:20:23 Universal 2017
[3|install_add]: START Sun Feb 26 05:55:31 UTC 2017
[3|install_add]: END SUCCESS Sun Feb 26 05:57:22 UTC 2017
[4|install_add]: END SUCCESS /bootflash/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Device# show install log
[0|install_op_boot]: START Fri Feb 24 19:20:19 Universal 2017
[0|install_op_boot]: END SUCCESS Fri Feb 24 19:20:23 Universal 2017
[3|install_add]: START Sun Feb 26 05:55:31 UTC 2017
[3|install_add]: END SUCCESS Sun Feb 26 05:57:22 UTC 2017
[4|install_add]: END SUCCESS /bootflash/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Device# show install log
[0|install_op_boot]: START Fri Feb 24 19:20:19 Universal 2017
[0|install_op_boot]: END SUCCESS Fri Feb 24 19:20:23 Universal 2017
[3|install_add]: START Sun Feb 26 05:55:31 UTC 2017
[3|install_add]: END SUCCESS Sun Feb 26 05:57:22 UTC 2017
[4|install_add]: END SUCCESS /bootflash/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Device# show install log
[0|install_op_boot]: START Fri Feb 24 19:20:19 Universal 2017
[0|install_op_boot]: END SUCCESS Fri Feb 24 19:20:23 Universal 2017
[3|install_add]: START Sun Feb 26 05:55:31 UTC 2017
[3|install_add]: END SUCCESS Sun Feb 26 05:57:22 UTC 2017
[4|install_add]: END SUCCESS /bootflash/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Device# show install log
[0|install_op_boot]: START Fri Feb 24 19:20:19 Universal 2017
[0|install_op_boot]: END SUCCESS Fri Feb 24 19:20:23 Universal 2017
[3|install_add]: START Sun Feb 26 05:55:31 UTC 2017
[3|install_add]: END SUCCESS Sun Feb 26 05:57:22 UTC 2017
[4|install_add]: END SUCCESS /bootflash/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
```

The table below lists the significant fields shown in the display.

**Table 1: show install summary Field Descriptions**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Packages</td>
<td>Name of the active model update package.</td>
</tr>
<tr>
<td>Inactive Packages</td>
<td>List of inactive packages.</td>
</tr>
<tr>
<td>Committed Packages</td>
<td>Installed model update packages that have saved or committed changes to the</td>
</tr>
<tr>
<td></td>
<td>hard disk, so that the changes become persistent across reloads.</td>
</tr>
</tbody>
</table>
Cisco Catalyst 3000 Series Switches

The following sample output from the `show install summary` command displays that the update package is now committed, and that it will be persistent across reloads:

```
Device# show install summary

Active Packages:
bootflash:cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin

Inactive Packages:
No packages

Committed Packages:
bootflash:cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin

Uncommitted Packages:
No packages

Device#
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncommitted Packages</td>
<td>Model update package activations that are non persistent.</td>
</tr>
</tbody>
</table>
show platform yang-management process

To display the status of the software processes required to support NETCONF-YANG, use the `show platform yang-management process` command in privileged EXEC mode.

```
show platform yang-management process [{monitor [{switch {switch-number |active |standby;} R0}]}{switch [{switch-number |active |standby}; R0]}
```

### Syntax Description

- **monitor**
  - (Optional) Displays detailed information about processes that are running.

- **switch switch-number**
  - (Optional) Displays information about the specified switch.

- **active**
  - (Optional) Displays information about the active instance of the switch.

- **standby**
  - (Optional) Displays information about the standby instance of the switch.

- **R0**
  - (Optional) Displays information about the Route Processor (RP) slot zero.

### Command Modes

Privileged EXEC (#)

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.3.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

The following is sample output from the `show platform software yang-management process` command:

```
Device# show platform software yang-management process

confd : Running
nssd : Running
syncfd : Running
ncsshd : Running
dmiauthd : Running
vtyserverutild : Running
opdatamgrd : Running
nginx : Running
ndbmand : Running
```

The table below lists the significant fields shown in the display.
Table 2: show platform software yang-management process Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>confd</td>
<td>Configuration daemon</td>
</tr>
<tr>
<td>nesd</td>
<td>Network element synchronizer daemon</td>
</tr>
<tr>
<td>syncfd</td>
<td>Sync from daemon</td>
</tr>
<tr>
<td>ncsshd</td>
<td>NETCONF Secure Shell (SSH) daemon</td>
</tr>
<tr>
<td>dmiauthd</td>
<td>Device management interfac e(DMI) authentication daemon</td>
</tr>
<tr>
<td>vtyserverutild</td>
<td>VTY server util daemon</td>
</tr>
<tr>
<td>opdatamgrd</td>
<td>Operational Data Manager daemon</td>
</tr>
<tr>
<td>nginx</td>
<td>NGINX web server</td>
</tr>
<tr>
<td>ndbmand</td>
<td>NETCONF database manager</td>
</tr>
</tbody>
</table>

The following is sample output from the `show platform software yang-management process monitor` command:

```
Device# show platform software yang-management process monitor

COMMAND PID S VSZ RSS %CPU %MEM ELAPSED
nginx 24689 S 139328 11996 0.0 0.2 24-02:00:55
nginx 24695 S 146544 6824 0.0 0.1 24-02:00:55
```

The table below lists the significant fields shown in the display.

Table 3: show platform software yang-management process monitor Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td>Command name</td>
</tr>
<tr>
<td>PID</td>
<td>Process ID</td>
</tr>
<tr>
<td>S</td>
<td>Process state</td>
</tr>
<tr>
<td>VSZ</td>
<td>Virtual memory size (in KB)</td>
</tr>
<tr>
<td>RSS</td>
<td>Resident set size (in KB)</td>
</tr>
<tr>
<td>%CPU</td>
<td>CPU usage percentage</td>
</tr>
<tr>
<td>%MEM</td>
<td>Memory usage percentage</td>
</tr>
<tr>
<td>ELAPSED</td>
<td>Elapsed execution time</td>
</tr>
</tbody>
</table>
show telemetry ietf subscription

To display information about telemetry subscriptions on a device, use the `show telemetry ietf subscription` command in user EXEC or privileged EXEC mode.

```
show telemetry ietf subscription {subscription-ID | all | configured | dynamic} [ {brief | detail} ]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subscription-ID</td>
<td>Subscription ID. Valid values are from 0 to 4294967295.</td>
</tr>
<tr>
<td>all</td>
<td>Displays all subscription information.</td>
</tr>
<tr>
<td>configured</td>
<td>Displays a list of subscriptions configured via CLI or NETCONF set config.</td>
</tr>
<tr>
<td>dynamic</td>
<td>Displays information about dynamic subscriptions created using the &lt;establish-subscription&gt; RPC.</td>
</tr>
<tr>
<td>brief</td>
<td>(Optional) Displays a brief summary of the subscription information.</td>
</tr>
<tr>
<td>detail</td>
<td>(Optional) Displays the subscription information in detail.</td>
</tr>
</tbody>
</table>

**Command Modes**

- User EXEC (>
- Privileged EXEC (#)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.6.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `show telemetry ietf subscription` command or the `<get>` RPC to retrieve the list of current subscription details on a device.

**Example**

The following is sample output from the `show telemetry ietf subscription subscription-ID detail` command:

```
Device# telemetry ietf subscription 2147483667 detail

Telemetry subscription detail:

  Subscription ID: 2147483667
  State: Valid
  Stream: yang-push
  Encoding: encode-xml
  Filter:
    Filter type: xpath
```
XPath: /mdt-oper:mdt-oper-data/mdt-subscriptions

Update policy:
   Update Trigger: periodic
   Period: 1000

Notes:

The following is sample output from the `show telemetry ietf subscription dynamic brief` command:

```
Device# show telemetry ietf subscription dynamic brief

Telemetry subscription brief

<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>State</th>
<th>Filter type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2147483667</td>
<td>Dynamic</td>
<td>Valid</td>
<td>xpath</td>
</tr>
<tr>
<td>2147483668</td>
<td>Dynamic</td>
<td>Valid</td>
<td>xpath</td>
</tr>
<tr>
<td>2147483669</td>
<td>Dynamic</td>
<td>Valid</td>
<td>xpath</td>
</tr>
</tbody>
</table>
```

The table below lists the significant fields shown in the display.

**Table 4: show telemetry ietf subscription Field Descriptions**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscription ID</td>
<td>Subscription identifier</td>
</tr>
<tr>
<td>State</td>
<td>Validity of a configured subscription. State will always be valid for dynamic subscriptions. For example, a configured subscription can be in a half-configured state, and therefore invalid. However, if a dynamic establish subscription is invalid, an error RPC response is sent back, and the subscription will not appear in this table.</td>
</tr>
<tr>
<td>Stream</td>
<td>Type of streaming used for subscriptions. Only YANG-push is supported.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Specifies encode-xml as the encoding type.</td>
</tr>
<tr>
<td>Filter Type</td>
<td>Type of filter used for subscriptions. Only XPath is supported.</td>
</tr>
<tr>
<td>XPath</td>
<td>XPath filter type or how the subscribed information was selected.</td>
</tr>
<tr>
<td>Update Trigger</td>
<td>Type of trigger to update subscriptions.</td>
</tr>
<tr>
<td>Period</td>
<td>Periodic timer configured to trigger an update. Values are specified in centiseconds (1/100 of a second).</td>
</tr>
<tr>
<td>Notes</td>
<td>A brief explanation about why a subscription is invalid. But for dynamic subscriptions, this field is always be empty.</td>
</tr>
</tbody>
</table>
### show telemetry ietf subscription

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Subscription ID.</td>
</tr>
</tbody>
</table>