Programmability

Programmability is supported only on Cisco Catalyst 4500-X Series Switches and Cisco Catalyst 4500E Series Switches with Supervisor Engine 9-E, 8L-E, and 8-E. The feature is supported on all available license levels for these switches. This chapter describes how to set-up and configure the feature. It includes the following major sections:

- About Programmability, page 6-1
- Configuring Programmability Components, page 6-4
- Using NETCONF and RESTCONF Protocols, page 6-14
- Using ODM Models, page 6-15
- Monitoring Programmability, page 6-23
- Troubleshooting Programmability, page 6-25
- Sample Configuration and Reference Information, page 6-28

About Programmability

- Overview, page 6-1
- Programmability Components, page 6-2
- Protocols and Data Models for Programmatic Device Configuration, page 6-2
- Default Configuration, page 6-3

Overview

Programmability is the capability to configure and manage networking devices using protocols that are specifically designed to be consumed by software, that is, machine to machine interfaces.

The traditional way of configuring and managing Cisco networking devices, has been manual configuration, through the command line interface (CLI). As deployments become more complex, programmability of devices has enabled a shift from manual to automatic network provisioning and configuration.

Managing device configuration programmatically enables you to:

- Configure and control at scale—You can automate network configuration while also overcoming difficulties posed by multiple platforms, multiple operating systems, and multiple vendor devices in your network.
Check to make sure that dependencies are satisfied before committing a change; and also easily roll-back changes when they are not consistently compatible across the network.

To address configuration and monitoring issues, the Internet Engineering Task Force (IETF) has defined these standards in network management:

- Yet Another Next Generation (YANG) data modeling—RFC 6020.
- Network Configuration Protocol (NETCONF)—RFC 6241

On Catalyst 4500 Series Switches, the Programmability feature provides the use of NETCONF and RESTCONF interfaces. They reside in a container on the switch and provide interfaces that enable remote management. The YANG data models available with these interfaces determine the scope of functions or actions that can be performed. See Figure 6-1.

### Programmability Components

This section describes the components involved in the setup of the feature. See the Configuring Programmability Components, page 6-4 for information about how to configure individual components.

- The Virtual Services Container—Also referred to as a virtual machine (VM), virtual service, or container, is a virtual environment on the switch. You can install an application within a virtual services container. The application then runs in the virtual services container of the operating system of a switch. The application is delivered as an open virtual application (OVA), which is a tar file with a .ova extension. The OVA package is installed and enabled on the switch through the device CLI.

- The Data Model Interface (DMI)—A container solution that provides the NETCONF and RESTCONF programmable interfaces. You must install and activate this container on the switch. After you activate it, the YANG models and APIs are available for use.

- The Network Bootloader—Network boot loaders support booting from a network-based source. On the Catalyst 4500 series switches, the Preboot Execution Environment (PXE) feature, also called PXE boot, enables the switch to retrieve the software image, configuration files, scripts, and ova files from a remote server, without end-user intervention, that is, Zero-Touch Provisioning. The remote server can be an HTTP or a TFTP server. PXE boot requires the configuration of a DHCP server, and the boot field set to one of the autoboot options in the ROMMON.

Throughout the document, PXE boot is used to refer to the method of booting from a network-based source.

### Protocols and Data Models for Programmatic Device Configuration

This section describes the protocols and modeling languages that enable a programmatic way of writing configurations to a network device.

- NETCONF—An XML-based protocol that you can use to request information from and make configuration changes to the switch. NETCONF Application Programming Interfaces (APIs) use Secure Shell Version 2 (SSHv2).
• RESTCONF—Uses structured data (XML or JSON) and YANG to provide a REST-like APIs, enabling you to programmatically access different network devices. RESTCONF APIs use HTTPS methods.

• YANG—A data modeling language that is used to model configuration and operational features on the switch. YANG determines the scope and the kind of functions that can be performed by NETCONF and RESTCONF APIs. The following data models are supported:
  – The `ned.yang` model—a Cisco-specific configuration data model that enables you to perform write (SET) operations.
  – The Operational Data Manager (ODM)—Enables you to read operational state data (GET operations) using YANG models.

Figure 6-1 shows how the different components of Programmability come together.

![Programmability Components Diagram]

**Default Configuration**

Programmability is not enabled.
Configuring Programmability Components

You can use Zero-Touch Provisioning to configure the programmability components or follow the standard configuration method (by configuring all required tasks individually).

The following applies to both methods of configuration:

- Prerequisites for Configuring Programmability, page 6-4
- Restrictions and Limitations for Configuring Programmability, page 6-5

For zero touch provisioning, you must ensure that you have met:

- Zero-Touch Provisioning Requirements, page 6-5

For the standard configuration method, you must complete the following:

- Installing and Activating the DMI Container, page 6-9
- Configuring One Platform Kit (OnePK), page 6-10
- Providing Privilege Access to Use NETCONF and RESTCONF, page 6-11
- Enabling the NETCONF Interface, page 6-12
- Enabling Cisco IOS HTTP Services for RESTCONF, page 6-13

Prerequisites for Configuring Programmability

- Prerequisites for NETCONF and RESTCONF:
  Your access to the switch is configured with privilege level 15. This is required to start working with NETCONF and RESTCONF interfaces. See Providing Privilege Access to Use NETCONF and RESTCONF, page 6-11.

- To use the programmability feature, you must use the Universal Crypto Image. See section “Orderable Product Numbers” in the corresponding release notes:
  - Release Notes for the Catalyst 4500-X Series Switches
  - Release Notes for the Catalyst 4500-E Series

- Prerequisites for using PXE boot:
  - The boot capability is set to autoboot and the bootfield is set to 04, 05, or 06. PXE boot is supported only if you have enabled autoboot.

  **Note** For PXE boot, the boot capability is set to autoboot by default.

  - The required ROMMON version is installed:
    On Catalyst 4500-X Series Switches, ROMMON version 15.0(1r)SG14 applies.
    On Catalyst 4500-E Series Switches, ROMMON version 15.1(1r)SG8 applies.
    With the above ROMMON versions, the factory default setting for the configuration register value is 0x2106 (boot field 06). This is also the recommended setting if you are using an existing device and upgrading to these ROMMON versions.

  **Note** If you are not using PXE boot, you do not have to upgrade the ROMMON version.
Restrictions and Limitations for Configuring Programmability

- Data model related restrictions:
  - Only a subset of the IETF, or common data models are supported.
  - Only the Cisco device-specific ned.yang model is supported.
  - When using ODM models, you must stop and restart the ODM control process if the crypto keys are regenerated. See Activating and Deactivating the ODM, page 6-17

- DMI solution related restrictions:
  - IPv6 is not supported.
  - Switches operating in the VSS mode are not supported.
  - It is not ISSU-capable.

- Only up to 4 simultaneous NETCONF sessions are supported. Further, a session that is idle for more than 180 seconds will timeout.

- Requests coming on an EtherChannel that is part of a Layer 3 Switched Virtual Interface (SVI), and is sharing its IP with the DMI container, is not supported

- AAA remote authentication is not supported.

Zero-Touch Provisioning Requirements

Zero-Touch Provisioning is achieved by using the PXE boot feature. Ensure that you have completed the following:

- Set the boot field value. See Boot Field, page 6-6
- Configured the DHCP server and an HTTP or TFTP server. See PXE Boot Requirements —Configuring the DHCP Server, page 6-6 and PXE Boot Process Flow, page 6-7
- Entered the following global configuration commands in the start-up configuration file. This refers to the <filename>.config file and is downloaded during the PXE boot process. This is required if you want NETCONF and RESTCONF to be available for use from Day 0.
  - The virtual-service DMI command (The virtual service name must be DMI if one opts for Zero-Touch Provisioning).
  - The activate command
  - The ip shared host-interface interface-id command
  - The onep command
  - The service set vty command
  - The username name privilege level password password command

To use NETCONF
  - The netconf-yang command

To use RESTCONF
  - The restconf command
  - The ip http server or the ip http secure-server command
  - The ip http authentication local command
The following is a sample output of the `show running-config` command. It displays the commands that are configured as part of start-up configuration:

```
Switch #show running-config
Building configuration...

<output truncated>
username dmi_admin privilege 15 password 0 dmi_admin
<output truncated>
interface GigabitEthernet3/47
  no switchport
ip address 198.51.100.20 255.255.255.0

<output truncated>
username admin privilege 15 password 7 070E25414707
line vty 0 4
login local
transport input all
ip http authentication local
ip http secure-server
onep
service set vty
restconf
netconf-yang
virtual-service DMI
activate
ip shared host-interface GigabitEthernet6/1
```

### Boot Field

To use PXE boot, you must enable autoboot, and set the boot field 04, 05, or 06. This automatically sets the corresponding configuration register value.

The PXE boot feature requires ROMMON version 15.0(1r)SG14 on Catalyst 4500-X Series Switches, and ROMMON version 15.1(1r)SG8 on Catalyst 4500-E Series Switches.

For detailed information about the various boot fields, see table *Explanation of Boot Field (Configuration Register Bits 00 to 03)* in chapter “Configuring the Switch for the First Time” in this book.

### PXE Boot Requirements —Configuring the DHCP Server

To send switch startup configuration files, scripts, and ova file in addition to the bootable image, you must configure the DHCP server.

Depending on your existing DHCP server setup (whether on Microsoft Windows or Linux), ensure that you have made the corresponding, requisite settings. See DHCP Server Settings on Linux, page 6-28 or Microsoft Windows DHCP Server Configuration, page 6-33, whichever applies.

---

**Note**  After completing DHCP server configuration, manually assign an IP to the switch and ping to check the switch-to-server connectivity.
Observe the following DHCP server configuration guidelines:

- You must provide the gateway, subnet mask, server IP address, and the client IP address. This information is not permanently stored on switch. They are used only to download files and are deleted when the activity is complete.

- Specifics for the start-up configuration file, script file and ova files (applies to DHCP server configuration for Microsoft Windows and Linux):
  - Complete the DHCP Option 43 list with information about the location of configuration, script and ova file to be downloaded.
  - You can specify an HTTP server or a TFTP server location from which to download. Depending on your requirement, specify one or more options—the boot image name, the start-up configuration file name and path, the script file name and path, and the ova file name and path. Ensure that the configuration, script and ova file extensions are 
    `<config-file>.config,<script-file>.script,<container-file>.ova` respectively.
  - If you are opting for Zero-Touch Provisioning, the ova file name must include _dmi_. For example, `example_dmi_container.ova`
  - These files should be saved in the root folder.
  - If you are using HTTP to download, you must configure the DNS server information.

- When the DHCP server responds successfully, the output displays `Received DHCP_ACK`.

- If you receive a TFTP timeout error, increase the DHCP timeout by using the ROMMON variable `DhcpTimeout`. The default DHCP timeout is 5 seconds. You can increase it by a maximum of 30 seconds. For example, if `DhcpTimeout=20`, the DHCP timeout increases by 20 seconds. Enter the set command to verify the change.
  ```
  rommon> DhcpTimeout=20
  rommon> set
  ```

- The DHCP options list that the DHCP server sends should not be greater than 255 bytes. If it is, the following error message is displayed:
  ```
  DHCP ERROR: Received Option length is more than maximum supported (255)
  ```

- The PXE process ignores network information that you configure on the ROMMON, such as IP, gateway, subnet mask etc.

- You can interrupt the autoboot process at any point, by pressing Control +C (switches to the ROMMON mode).
  - For sample output of the autoboot process, using HTTP, see Autoboot Process Output—Using HTTP, page 6-37.
  - For sample output of the autoboot process using TFTP, see Autoboot Process Output—Using TFTP, page 6-40

**PXE Boot Process Flow**

This section outlines the communication process between the DHCP server and the switch and provides the sequence of events followed during the PXE boot (network boot loading) process. This assumes that autoboot is enabled.

1. The switch sends a DHCP discovery packet.
2. The DHCP server responds with an offer containing the TFTP or HTTP server IP address, the offered IP address for the client, the gateway IP address, the boot file name, and the path and names of the OVA, script, and switch configuration files.
3. The switch sends the DHCP request for the IP address.

4. The switch receives the DHCP acknowledgment packet from the server, downloads the image specified in the `filename` variable in the DHCP server, and then boots up with downloaded image.

   While booting, the switch receives the Option 43 list from the DHCP server with information about the location of configuration file, script file and ova file to be downloaded.

5. After POST is complete, the switch looks for the startup configuration files, script files, and ova files as mentioned in the Option 43 list received in Step 4. If the files mentioned the Option 43 list are present in the specified location, the switch downloads them.

   The script file is downloaded to— bootflash:pxe/scripts folder
   The ova file is downloaded to— bootflash:pxe/ova folder.

---

**Figure 6-2  PXE Boot Process Flow**

* DNS Server steps do not apply if you are using a TFTP Server
## Installing and Activating the DMI Container

This task is mandatory if you have opted for the standard configuration method.

Before you begin, ensure that you have completed the following:

- Downloaded an OVA package that is compatible with the device operating system. The OVA package is available for download in the same location as your system image (.bin) file.
- Ensured that the minimum required disk space - 512 MB, and memory - 256 MB RAM is available on the device for installation and deployment of the DMI container.

To install and activate the DMI by using the virtual services container CLI, perform the following task:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>enable</td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Switch# enable</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>virtual-service install name virtual-services-name package file</td>
<td>Installs an OVA package from the specified location onto a device. Ensure that the ova file is located in the root directory of the storage device.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Switch# virtual-service install name dmi package bootflash:/dmi.ova</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>[no] virtual-service virtual-services-name</td>
<td>Configures a virtual services container and enters virtual services configuration mode. Observe these guidelines:</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Switch(config)# virtual-service dmi</td>
<td>• Use the virtual-services-name defined during installation of the application.</td>
</tr>
<tr>
<td></td>
<td>Switch(config-virt-serv)#</td>
<td>• Ensure that installation is complete before proceeding to the next step using the show virtual-service list command.</td>
</tr>
<tr>
<td>5</td>
<td>[no] activate</td>
<td>Activates the installed virtual services container.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Switch(config-virt-serv)# activate</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Programmability Components

Chapter 6      Programmability

Configuring One Platform Kit (OnePK)

OnePK is a software development kit. It enables you to create applications with which to interact directly with Cisco networking devices, and also use a set of controlled API to access networking services.

In the context of Programmability, it is used to enable the VTY service set. The VTY Service allows a OnePK application to communicate with a network element, through a virtual terminal.

This task is mandatory if you have opted for the standard configuration method. To enable the requisite, internal OnePK infrastructure, perform the following task:

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td>Example: Switch&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td>Example: Switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 onep</td>
<td>Enters the OneP configuration mode.</td>
</tr>
<tr>
<td>Example: Switch(config)# onep Switch(config-onep)#</td>
<td></td>
</tr>
</tbody>
</table>
Providing Privilege Access to Use NETCONF and RESTCONF

This task is mandatory for both zero touch provisioning, and the standard configuration method. To start working with NETCONF and RESTCONF APIs you must be a user with privilege level 15. To provide this, perform the following task:

Command or Action | Purpose
--- | ---
Step 1 enable | Enables the privileged EXEC mode. Enter your password if prompted.
Example: Switch# enable
Step 2 configure terminal | Enters the global configuration mode.
Example: Switch# configure terminal
Step 3 username name privilege level password password
Example: Switch (config)# username example-name privilege 15 password example_password
 | Establishes a user name-based authentication system. Configure the following keywords:
• privilege level—Sets the privilege level for the user. For the programmability feature, it must be 15.
• password password—Sets a password to access the CLI view.
Step 4 end | Exits global configuration mode and enters privileged EXEC mode.
Example: Switch# end
Enabling the NETCONF Interface

This task is mandatory if you want to use the NETCONF interface and have opted for the standard configuration method:

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td>Example: Switch# enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td>Example: Switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 netconf-yang</td>
<td>Enables the NETCONF interface on your network device. After you have completed this step, you can manage network devices through a model based interface. The complete activation of model-based interface processes may require up to 90 seconds.</td>
</tr>
<tr>
<td>Example: Switch(config)# netconf-yang</td>
<td></td>
</tr>
<tr>
<td>Step 4 end</td>
<td>Exits global configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Switch# end</td>
<td></td>
</tr>
</tbody>
</table>

See Examples for NETCONF RPCs, page 6-14.
Enabling Cisco IOS HTTP Services for RESTCONF

This task is mandatory if you want to use the RESTCONF interface and have opted for the standard configuration method.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Switch# enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 restconf</td>
<td>Enables the RESTCONF interface on your network device.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Switch(config)# restconf</td>
<td></td>
</tr>
<tr>
<td>Step 4 ip http server</td>
<td>The ip http server command enables the HTTP server on your system.</td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>ip http secure-server</td>
<td>The ip http secure-server command enables a secure HTTP (HTTPS) server.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Switch (config)# ip http server</td>
<td></td>
</tr>
<tr>
<td>OR</td>
<td></td>
</tr>
<tr>
<td>Switch (config)# ip http secure-server</td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td>When enabling an HTTPS server, you should always disable the standard HTTP server to prevent unsecured connections to the same services. Disable the standard HTTP server using the no ip http server command in global configuration mode (this step is precautionary; typically, the HTTP server is disabled by default). Configure only one of the commands.</td>
</tr>
<tr>
<td>Step 5 ip http authentication local</td>
<td>Indicates that the login user name, password and privilege level access combination specified in the local system configuration (with the username global configuration command) should be used for authentication and authorization.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Switch(config)# ip http authentication local</td>
<td></td>
</tr>
<tr>
<td>Step 6 end</td>
<td>Exits global configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Switch# end</td>
<td></td>
</tr>
</tbody>
</table>

See Examples for RESTCONF RPCs, page 6-15.
Using NETCONF and RESTCONF Protocols

NETCONF uses a simple RPC-based (Remote Procedure Call) mechanism to facilitate communication between a client and a server. The client can be a script or an application running as part of a network manager. The server is typically a network device (switch or router).

NETCONF uses Secure Shell Version 2 (SSHv2) as the transport layer across network devices and RESTCONF uses HTTP.

To use NETCONF and RESTCONF you must complete all the required tasks as per the Configuring Programmability Components, page 6-4 section.

NETCONF and RESTCONF also support capability discovery and model downloads. Supported models are discovered using the ietf-netconf-monitoring model. Revision dates for each model are shown in the capabilities response. Data models are available for optional download from a device using the get-schema rpc. You can use these YANG models to understand or export the data model.

The following shows sample RPCs you can send and the kind of action that is performed.

- Examples for NETCONF RPCs, page 6-14
- Examples for RESTCONF RPCs, page 6-15

Examples for NETCONF RPCs

Get the running-configuration of the switch by sending the following RPC:

```xml
<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"/>
    </filter>
  </get-config>
</rpc>
```

Change the description of an interface by sending the following RPC:

```xml
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <native xmlns="http://cisco.com/ns/yang/ned/ios">
        <interface>
          <TenGigabitEthernet>
            <name>4/1</name>
            <description>to_distribution</description>
          </TenGigabitEthernet>
        </interface>
      </native>
    </config>
  </edit-config>
</rpc>
```
Remove the description from an interface by sending the following RPC

```xml
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <native xmlns="http://cisco.com/ns/yang/ned/ios">
        <interface>
          <TenGigabitEthernet>
            <name>4/1</name>
            <description xc:operation="delete"/>
          </TenGigabitEthernet>
        </interface>
      </native>
    </config>
  </edit-config>
</rpc>
```

Examples for RESTCONF RPCs

Get the TFTP source interface by sending the following RPC:

```
```

Configure the TFTP source interface by sending the following RPC:

```
payload = "{"GigabitEthernet": "2/2"}"
```

Enter an HTTP delete request by sending the following RPC:

```
```

For the HTTP delete request do not use:

```
```

Using ODM Models

You use ODM models to retrieve read-only operational state data from the system. For this, you must enable Secure Shell Version 2 (SSHv2), activate the ODM, and set the time interval at which the models will collect information from the system.

Each ODM model has a corresponding parser, which polls the specified operational data according to the specified polling interval in milliseconds. See the following sections:

- **Enabling SSHv2**, page 6-17
- **Activating and Deactivating the ODM**, page 6-17
- **Enabling the Polling Mode**, page 6-19
- **Displaying Supported Parsers and Polling Intervals**, page 6-20
The following tables lists the parsers, ODM models, and the kind of operational state data that is polled. By default, polling is enabled.

<table>
<thead>
<tr>
<th>No.</th>
<th>Parsers</th>
<th>ODM Models</th>
<th>Corresponding show Command and Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>parse.showArchive</td>
<td>cisco-checkpoint-archive.yang</td>
<td>Corresponds to the <code>show archive</code> command, which displays information about the files saved in the Cisco IOS configuration archive.</td>
</tr>
<tr>
<td>2</td>
<td>parse.showACL</td>
<td>cisco-acl-oper.yang (confirm if a footnote is required - pratyusha)</td>
<td>Corresponds to the <code>show ip access-lists</code> command, which displays the contents of all current IP access lists.</td>
</tr>
<tr>
<td>3</td>
<td>parse.showVirtualService</td>
<td>cisco-virtual-service.yang</td>
<td>Corresponds to the <code>show virtual-service list</code> command, which displays an overview of resources utilized by the applications</td>
</tr>
<tr>
<td>4</td>
<td>parse.showProcessesMemory</td>
<td>cisco-process-memory.yang</td>
<td>Corresponds to the <code>show processes memory</code> command, which displays the amount of memory used by each system process in Cisco IOS, Cisco IOS XE, or Cisco IOS Software Modularity images.</td>
</tr>
<tr>
<td>5</td>
<td>parse.showProcessesCPU</td>
<td>cisco-process-cpu.yang</td>
<td>Corresponds to the <code>show processes cpu</code> command, which displays CPU utilization to identify the causes of high CPU utilization.</td>
</tr>
<tr>
<td>6</td>
<td>parse.showIpRoute</td>
<td>ietf-routing.yang</td>
<td>Corresponds to the <code>show ip route</code> command, which displays the current state of the routing table to verify the configuration.</td>
</tr>
<tr>
<td>7</td>
<td>parse.showInterfaces</td>
<td>ietf-interfaces.yang</td>
<td>Corresponds to the <code>show interfaces</code> command, which displays statistics for all interfaces configured on the device or access server.</td>
</tr>
<tr>
<td>8</td>
<td>parse.showBFDneighbors</td>
<td>cisco-bfd-state.yang</td>
<td>Corresponds to the <code>show bfd neighbors</code> command, which displays the active BFD neighbor and displays the routing protocols that BFD has registered.</td>
</tr>
<tr>
<td>9</td>
<td>parse.showLLDPneighbors</td>
<td>cisco-lldp-state.yang</td>
<td>Corresponds to the <code>show lldp neighbors</code> command, which displays information about neighbors, including device type, interface type and number, holdtime settings, capabilities, and port ID.</td>
</tr>
<tr>
<td>10</td>
<td>parse.showMacAddTable</td>
<td>cisco-mac-address-table.yang</td>
<td>Corresponds to the <code>show mac-address-table</code> command, which displays the MAC address table.</td>
</tr>
<tr>
<td>11</td>
<td>parse.showPower</td>
<td>cisco-poe-interfaces.yang</td>
<td>Corresponds to the <code>show power inline</code> command, which displays the PoE state for the switch.</td>
</tr>
<tr>
<td>12</td>
<td>parse.showModule</td>
<td>cisco-equipment-module.yang</td>
<td>Corresponds to the <code>show module</code> command, which displays module status.</td>
</tr>
<tr>
<td>13</td>
<td>parse.showVersion</td>
<td>cisco-cat4k-version.yang</td>
<td>Corresponds to the <code>show version</code> command, which displays hardware and software information for the system.</td>
</tr>
</tbody>
</table>
Chapter 6  Programmability

Using ODM Models

### Enabling SSHv2


### Activating and Deactivating the ODM

This section contains sample RPCs to check the current status of the ODM, to activate it, and to deactivate it.

#### Example: Checking the Current Status of the ODM.

In the example below, the ODM is active (value set to true).

**Input**

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

**Output**

```xml
NETCONF RETURN
---
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <data>
    ...
  </data>
</rpc-reply>
```

---

<table>
<thead>
<tr>
<th>No.</th>
<th>Parsers</th>
<th>ODM Models</th>
<th>Corresponding show Command and Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>parse.showInventory</td>
<td>cisco-inventory-entities.yang</td>
<td>Corresponds to the <code>show inventory</code> command, which displays product identification (PID) information for the hardware</td>
</tr>
<tr>
<td>15</td>
<td>parse.showIntTransciver</td>
<td>cisco-interface-transciver.yang</td>
<td>Corresponds to the <code>show interfaces transceiver detail</code> command, which displays information about the optical transceivers that have digital optical monitoring (DOM) enabled.</td>
</tr>
<tr>
<td>16</td>
<td>parse.showIgmpGroup</td>
<td>cisco-igmpsn-group.yang</td>
<td>Corresponds to the <code>show ip igmp snooping groups</code> command, which displays the member port and the IP address.</td>
</tr>
<tr>
<td>17</td>
<td>parse.showFlowMonitor</td>
<td>cisco-flow-monitor.yang</td>
<td>Corresponds to the <code>show flow monitor name cache</code> command, which displays the status and statistics for a Flexible NetFlow flow monitor.</td>
</tr>
<tr>
<td>18</td>
<td>showIPslaStatistics</td>
<td>cisco-ip-sla-stats.yang</td>
<td>Corresponds to the show <code>ip sla statistics</code> command, which displays the current operational status and statistics of all IP SLAs operations or a specified operation.</td>
</tr>
</tbody>
</table>
Example: Activating or Starting the ODM

In the example below, the RPC reply (ok) indicates that the ODM is activated successfully.

**Input**

```xml
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <cisco-ia xmlns="http://cisco.com/yang/cisco-ia">
        <odm-control>true</odm-control>
      </cisco-ia>
    </config>
  </edit-config>
</rpc>
```

**Output**

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

To deactivate or stop the ODM, send the following RPC:

**Input**

```xml
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <cisco-ia xmlns="http://cisco.com/yang/cisco-ia">
        <odm-control>false</odm-control>
      </cisco-ia>
    </config>
  </edit-config>
</rpc>
```

**Output**

```
NETCONF RETURN
------------
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <ok/>
</rpc-reply>
```
Enabling the Polling Mode

This section contains sample RPCs to check the current polling mode, to enable or change it, and to change the polling interval.

Example: Verifying the Currently Polling Mode of the ODM Models

In the example below, polling is enabled. (In the output section, the polling-enable parameter is set to true).

Input

```xml
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <cisco-odm xmlns="http://cisco.com/yang/cisco-odm">
        <polling-enable/>
      </cisco-odm>
    </filter>
  </get>
</rpc>
```

Output

NETCONF RETURN

```xml
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <data>
    <cisco-odm xmlns="http://cisco.com/yang/cisco-odm">
      <polling-enable>true</polling-enable>
    </cisco-odm>
  </data>
</rpc-reply>
```

Example: Enabling or Changing the Polling Mode of the ODM Models

Input

```xml
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <cisco-odm xmlns="http://cisco.com/yang/cisco-odm">
        <polling-enable>true</polling-enable>
      </cisco-odm>
    </config>
  </edit-config>
</rpc>
```

Output

NETCONF RETURN

```xml
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <ok/>
</rpc-reply>
```
Example: Changing the Polling Interval of a Parser

In the example below, the polling interval of parser `parse.showArchive` is changed to 110000 milliseconds:

**Input**

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <cisco-odm xmlns="http://cisco.com/yang/cisco-odm">
        <actions>
          <action-name>parse.showArchive</action-name>
          <polling-interval>110000</polling-interval>
        </actions>
      </cisco-odm>
    </config>
  </edit-config>
</rpc>
```

**Output**

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

Displaying Supported Parsers and Polling Intervals

To retrieve information about all the supported parsers and their polling intervals, send the following RPC:

**Input**

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <cisco-odm xmlns="http://cisco.com/yang/cisco-odm">
        <actions>
          <action-name/>
          <polling-interval/>
          <mode/>
        </actions>
      </cisco-odm>
    </filter>
  </get>
</rpc>
```

**Output**

```
NETCONF RETURN
--------------
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <data>
  </data>
</rpc-reply>
```
<cisco-odm xmlns="http://cisco.com/yang/cisco-odm">
    <actions>
        <action-name>parse.showACL</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showArchive</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showBFDneighbors</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showFlowMonitor</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showIgmpGroup</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showIntTransciver</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showInterfaces</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showInventory</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showIpRoute</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showLLDPneighbors</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showMacAddTable</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showModule</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
</cisco-odm>
<mode>poll</mode>
</actions>
</mode>
</rpc-reply>

Ran 1 test in 0.583s

OK
Monitoring Programmability

Use these commands in the privileged EXEC mode, to display the Programmability settings you have configured:

Table 6-1 Monitoring Programmability

<table>
<thead>
<tr>
<th>Show Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>debug remotemanagement dmi</td>
<td>Displays the list of applications for which you can debug information. You can choose from the following list of applications:</td>
</tr>
<tr>
<td></td>
<td>• ciaauthd—Debug CIA Authorization Daemon</td>
</tr>
<tr>
<td></td>
<td>• ciam—Debug CiaManager</td>
</tr>
<tr>
<td></td>
<td>• confd—Debug Confd</td>
</tr>
<tr>
<td></td>
<td>• monit—Debug Monit</td>
</tr>
<tr>
<td></td>
<td>• nes—Debug Network Element Synchronizer</td>
</tr>
<tr>
<td></td>
<td>• odm—Debug Operational Data Manager</td>
</tr>
<tr>
<td></td>
<td>• snmp—Debug SNMP Notification Processor</td>
</tr>
<tr>
<td></td>
<td>• syncfd—Debug SyncFromDaemon</td>
</tr>
<tr>
<td>show onep session all</td>
<td>Displays OneP session information. To verify if NETCONF and RESTCONF interfaces are configured correctly, ensure that these three sessions are listed: NetworkElementSynchronizer, SyncFromDaemon and CiaAuthDaemon. The following is sample output for this command:</td>
</tr>
<tr>
<td></td>
<td>Switch # show onep session all</td>
</tr>
<tr>
<td></td>
<td>ID Username State ReconnectTimer ConnectTime ApplicationName</td>
</tr>
<tr>
<td></td>
<td>8145 Connected 0 Thu Jul 28 06:07:05.304 com.cisco.NetworkElementSynchronizer</td>
</tr>
<tr>
<td></td>
<td>3234 Connected 0 Thu Jul 28 06:07:06.504 com.cisco.SyncFromDaemon</td>
</tr>
<tr>
<td></td>
<td>7249 Connected 0 Thu Jul 28 06:07:07.343 com.cisco.CiaAuthDaemon</td>
</tr>
<tr>
<td>show remotemanagement dmi</td>
<td>Displays the list of applications for which you can display log and status information. You can choose from the following list of applications:</td>
</tr>
<tr>
<td></td>
<td>• ciaauthd—Show CIA Authorization Daemon</td>
</tr>
<tr>
<td></td>
<td>• ciam—Show CiaManager</td>
</tr>
<tr>
<td></td>
<td>• confd—Show Confd</td>
</tr>
<tr>
<td></td>
<td>• genet—Show Mapping Framework</td>
</tr>
<tr>
<td></td>
<td>• log—Show all DMI logs</td>
</tr>
<tr>
<td></td>
<td>• monit—Show Monit</td>
</tr>
<tr>
<td></td>
<td>• nes—Show Network Element Synchronizer</td>
</tr>
<tr>
<td></td>
<td>• odm—Show Operational Data Manager</td>
</tr>
<tr>
<td></td>
<td>• snmp—Show SNMP Notification Processor</td>
</tr>
<tr>
<td></td>
<td>• status—Show status for all DMI applications</td>
</tr>
<tr>
<td></td>
<td>• syncfd—Show SyncFromDaemon</td>
</tr>
<tr>
<td>show virtual-service [global]</td>
<td>Displays available memory, disk space, and CPU allocated for applications.</td>
</tr>
</tbody>
</table>
### Table 6-1 Monitoring Programmability

<table>
<thead>
<tr>
<th>Show Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show virtual-service detail [name \n virtual-services-name]</td>
<td>Displays a list of resources committed to a specified application, including attached devices.</td>
</tr>
<tr>
<td>show virtual-service list</td>
<td>Displays the list of applications installed in the virtual services container. The following is sample output for this command:</td>
</tr>
<tr>
<td></td>
<td>Switch# show virtual-service list</td>
</tr>
<tr>
<td></td>
<td>Virtual Service List:</td>
</tr>
<tr>
<td></td>
<td>Name Status Package Name</td>
</tr>
<tr>
<td></td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>dmi Activated cat4500e_20160725-212823.ova</td>
</tr>
<tr>
<td>show virtual-service storage pool list</td>
<td>Displays an overview of storage locations (pools) used for virtual service containers.</td>
</tr>
<tr>
<td>show virtual-service storage volume list</td>
<td>Displays an overview of storage volume information for virtual service containers.</td>
</tr>
<tr>
<td>show virtual-service version name \n virtual-services-name installed</td>
<td>Displays the version of an installed application. For example:</td>
</tr>
<tr>
<td></td>
<td>Switch# show virtual-service version name dmi installed</td>
</tr>
<tr>
<td></td>
<td>Virtual service dmi installed version:</td>
</tr>
<tr>
<td></td>
<td>Name : Netconf-Yang</td>
</tr>
<tr>
<td></td>
<td>Version : 1.0.0</td>
</tr>
<tr>
<td>show virtual-service tech-support</td>
<td>Displays container-based information.</td>
</tr>
<tr>
<td>show virtual-service redundancy state</td>
<td>Displays synchronization status</td>
</tr>
<tr>
<td>show virtual-service utilization statistics CPU</td>
<td>Displays virtual service CPU utilization statistics.</td>
</tr>
</tbody>
</table>
Troubleshooting Programmability

This section shows sample output for the some of the errors you may encounter while configuring the feature. In some cases a solution is described, and in others, sample configuration output serves as a guideline for correct configuration.

- File Not Found Errors, page 6-25
- Startup Configuration Errors, page 6-27
- Debugging the DMI, page 6-27

File Not Found Errors

If you receive such an error, check the path you have entered for the filename field in the DHCP configuration file and make sure that the file exists in your TFTP server. The sample output below shows a successful TFTP session:

Filename: /cat4500e-universalk9.SSA.03.09.00.PR4.46.152-5.0.46.PR4.bin
IP Address: 192.168.20.16
Loading from TftpServer: 10.106.24.187
TftpBlkSize: 1468
RxDataPacket: 130207

Loaded 191143008 bytes successfully.

Checking digital signature...
[/cat4500e-universalk9.SSA.03.09.00.PR4.46.152-5.0.46.PR4.bin]
Digitally Signed Development Software with key version A

Rommon reg: 0x00084F80
Reset2Reg: 0x00004F00

Image load status: 0x00000000
###
Winter 110 controller 0x0468AFAC..0x047F4313 Size:0x002FDB9D
Program Done!

Starting System Services
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=4,mode=600,ptmxmode=000 0 0
diagsk10-post version 5.1.4.1

diagsk10-post version 5.1.4.1
prod: WS-C4500X-16 part: 73-13860-03 serial: JAE155209ZG

Power-on-self-test for Module 1: WS-C4500X-16

CPU Subsystem Tests ...
seeprom: Pass
Traffic: L3 Loopback ...
Test Results: Pass
Traffic: L2 Loopback ...
Test Results: Pass
Exiting to ios...

Downloading config files from 10.106.24.187 to /bootflash/pxe/user-startup-config
configs/4500x_start.config
.Received 2201 bytes in 0.0 seconds
Downloading script files from 10.106.24.187 to /bootflash/pxe/scripts
scripts/hello.script
.Received 90 bytes in 0.0 seconds
Downloading ova files from 10.106.24.187 to /bootflash/pxe/ova
container/cat4500e_20160717-183651_33.ova
...............Received 164270080 bytes in 32.0 seconds
Continuing with IOS boot..
Aug 1 06:23:42 %IOSXE-3-PLATFORM: process kernel: [ 124.746012]
mpc85xx_pci_err_probe: Unable to request irq 0 for MPC85xx PCI err
Aug 1 06:23:42 %IOSXE-3-PLATFORM: process kernel: [ 124.756621]
mpc85xx_pci_err_probe: Unable to request irq 0 for MPC85xx PCIE err
Loading gsb64atomic as gdb64atomic
Loading pds_helper module
Loading container module
Failed to bring interface "eth1" up
Using 1 for MTS slot
Platform Manager: starting in standalone mode (active)

Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3 Switch Software
(cat4500e-UNIVERSALK9-M), Version 03.09.00.PR4.46 EARLY DEPLOYMENT [PROD IMAGE]
ENGINEERING NOVA WEEKLY BUILD, synced to V152_5_1_E
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Sun 31-Jul-16 16:31 by sabind

Cisco IOS-XE software, Copyright (c) 2005-2015 by cisco Systems, Inc.
All rights reserved. Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0. The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.
(http://www.gnu.org/licenses/gpl-2.0.html) For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.

This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:

If you require further assistance please contact us by sending email to
export@cisco.com.

cisco WS-C4500X-16 (MPC8572) processor (revision 3) with 4194304K bytes of physical memory.
Processor board ID JAE155209ZG
MPC8572 CPU at 1.5GHz, Cisco Catalyst 4500X
Last reset from Reload
1 Virtual Ethernet interface
16 Ten Gigabit Ethernet interfaces
511K bytes of non-volatile configuration memory.

Press RETURN to get started!
Switch>

Startup Configuration Errors
If you encounter errors when you replace the existing startup configuration with a new configuration, the system does not replace the existing startup configuration. You must resolve the errors in the switch start-up configuration file before resuming.

Debugging the DMI
To start debugging the DMI container, perform the following task:

Step 1 Set the logging level to “debug” in cisco-ia.yang model.

Step 2 In the privilege EXEC Mode on the switch CLI, enter one of these commands and use NETCONF to read the responses
   – The debug remotemanagement command.
   – The show remotemanagement command

Step 3 To display NETCONF statistical information, such as, the number of sessions, netconf RPCs, packets and so on, use the ietf-netconf-monitoring.yang model.

The following is sample output for the show remotemanagement command confd log command:

Switch# show remotemanagement dmi confd log
remotemanagement-iosxe-remote-mgmt-8086

***************************************************************************
cia-confd.log ***************************************************************************
<DEBU3> 16-Nov-2016::22:58:44.010 iosxe-remote-mgmt confd[28320]: - Loading file ./mib-fxs/VPN-TC-STD-MIB.fxs
<DEBU3> 16-Nov-2016::22:58:44.018 iosxe-remote-mgmt confd[28320]: - Loading file ./mib-fxs/IANA-ADDRESS-FAMILY-NUMBERS-MIB.fxs
<DEBU3> 16-Nov-2016::22:58:44.019 iosxe-remote-mgmt confd[28320]: - Loading file ./mib-fxs/IANA-ADDRESS-FAMILY-NUMBERS-MIB.fxs
<DEBU3> 16-Nov-2016::22:58:44.020 iosxe-remote-mgmt confd[28320]: - Loading file ./mib-fxs/IEEE8021-TC-MIB.fxs
<DEBU3> 16-Nov-2016::22:58:44.020 iosxe-remote-mgmt confd[28320]: - Loading file ./mib-fxs/IEEE8021-TC-MIB.fxs
Sample Configuration and Reference Information

This chapter provides sample configuration for the following:

- DHCP Server Settings on Linux, page 6-28
- Configuring DHCP Option 43 (for Microsoft Windows), page 6-32
- Microsoft Windows DHCP Server Configuration, page 6-33
- Autoboot Process Output, page 6-37

DHCP Server Settings on Linux

The following is sample configuration that is saved in the `dhcpd.conf` file. Use it as a point of reference when you configure DHCP server settings on Linux.

This sample output covers a scenario where different files are sent to multiple devices of the same vendor specific class, but each one of the devices has a different MAC address.

Comments throughout the sample configuration provide guidelines for important steps (sentence starting with #).

Note: You must restart the DHCP service every time you make a change in the `dhcpd.conf` file.

Depending on whether you are using HTTP or TFTP to download files, refer to the corresponding sample configuration file:

- Using HTTP, page 6-28
- Using TFTP, page 6-30

Using HTTP

```bash
allow booting;
allow bootp;
ddns-update-style none;
#DEFINE AN OPTION SPACE. "EXAMPLE" IS USED HERE. IT IS A VARIABLE YOU CAN SET.
#MAINTAIN code 1,2 AND 3 CONSISTENTLY SINCE THE VALUES CORRESPOND TO CONFIG,SCRIPT AND
#OVA FILES RESPECTIVELY.
option space EXAMPLE;
option EXAMPLE.startup-config code 1=text;
option EXAMPLE.user-script code 2=text;
option EXAMPLE.user-ova code 3=text;

#ENTER THESE DETAILS AS APPLICABLE TO YOUR NETWORK. TO DOWNLOAD USING HTTP, ENTER THE
#DNS SERVER DETAILS
option domain-name "example-httpserver-url.com";
option domain-name-servers 198.51.100.3;

option subnet-mask 255.255.255.0;
option broadcast-address 192.0.2.255;

#DEFINE A CLASS FOR THE VENDOR-SPECIFIC IDENTIFIER NAME THAT THE DEVICE HAS.
#EXAMPLE: FOR SUP8E/8LE IT IS "WS-X45-SUP8L-E"
#FOR CATALYST 4500-X IT IS "WS-4500X-16"
#ALSO DEFINE THE ROUTER,HTTP SERVER IDENTIFIER,NEXT SERVER IP DETAILS - AS APPLICABLE
#TO YOUR NETWORK
```
class "WS-X45-SUP8L-E" {
    match pick-first-value (option dhcp-client-identifier, hardware);
    option routers 198.0.2.254;
    option subnet-mask 255.255.255.0;
    server-identifier 198.51.100.2;
    next-server 198.51.100.2;
}

class "WS-4500X-16" {
    match pick-first-value (option dhcp-client-identifier, hardware);
    option routers 198.0.2.254;
    option subnet-mask 255.255.255.0;
    server-identifier 198.51.100.2;
    next-server 198.51.100.2;
}

#DEFINE A SUBCLASS TO ADD THE DEVICE BASED ON ITS MAC ADDRESS TO RECEIVE
#CONFIGURATION FILES. THIS APPLIES WHEN YOU HAVE MULTIPLE DEVICES WITH SAME
#VENDOR-SPECIFIC IDENTIFIER AND YOU WANT TO PUSH DIFFERENT CONFIGURATIONS BASED ON THE
#MAC ADDRESS

    # ENTER THE BOOTFILENAME.THE .BIN IMAGE FILE SHOULD RESIDE IN THE TFTPBOOT FOLDER.
    filename "example-ios-image.bin";
    option routers 198.0.2.254;
    #SPECIFY THAT THE OPTION 43 AND ROUTER(3) DETAILS HAVE TO BE SENT TO THE CLIENT SWITCH
    option dhcp-parameter-request-list 43,3;
    option vendor-class-identifier "WS-X45-SUP8L-E";
    vendor-option-space EXAMPLE;
    #SPECIFY THE PATH OF THE FILES YOU WANT TO SEND (HTTP).
    #MAKE SURE THESE FILES RESIDE IN IDENTICAL FOLDERS (configs/,scripts/,container/) IN
    #the HTTP ROOT FOLDER. YOU MUST CREATE THE IDENTICAL FOLDERS WITH THE SAME NAME AND
    #CASE.
    #ENTER A FILE NAME. MAKE SURE THAT CONFIG, SCRIPT, AND OVA FILE EXTENSIONS ARE
    #<config-file>.config,<script-file>.script,<container-file>.ova RESPECTIVELY.
    option EXAMPLE.startup-config
    "http://example-httpservr-url.com/configs/example-config.config";
    option EXAMPLE.user-script
    "http://example-httpservr-url.com/scripts/example-script.py";
    option EXAMPLE.user-ova
    "http://example-httpservr-url.com/container/example_dmi_container.ova";
    option dhcp-parameter-request-list 43,3;
}

subclass "WS-X45-SUP8L-E" 1:e4:aa:5d:c4:a5:a1 {
    #WHEN USING HTTP TO DOWNLOAD FILES, PROVIDE THE PATH IN THE FOLLOWING FORMAT:
    #filename "http://<http server url>/ios_image.bin"
    filename "http://example-httpservr-url.com/example-ios-image.bin"
    option routers 198.0.2.254;
    option dhcp-parameter-request-list 43,3;
    option vendor-class-identifier "WS-X45-SUP8L-E";
    vendor-option-space EXAMPLE;
    option EXAMPLE.startup-config
    "http://example-httpservr-url.com/example-config.config";
    option EXAMPLE.user-script
    "http://example-httpservr-url.com/example-script.py";
    option EXAMPLE.user-ova
    "http://example-httpservr-url.com/example-container.ova";
    option dhcp-parameter-request-list 43,3;


```plaintext
subclass "WS-4500X-16" 1:30:e4:db:f8:a4:9f {
    filename "example-ios-image.bin";
    option routers 198.0.2.254;
    option dhcp-parameter-request-list 43,3;
    option vendor-class-identifier "WS-4500X-16";
    vendor-option-space EXAMPLE;
    option EXAMPLE.startup-config
        "http://example-httpserver-url.com/example-config.config";
    option EXAMPLE.user-script
        "http://example-httpserver-url.com/example-script.py";
    option EXAMPLE.user-ova
        "http://example-httpserver-url.com/example-container.ova";
    option dhcp-parameter-request-list 43,3;
}

#ASSIGN A POOL TO GIVE IP ADDRESSES TO THE MEMBERS OF THE VENDOR-SPECIFIC CLASS
subnet 192.0.2.0 netmask 255.255.255.0 {
    pool {
        allow members of "WS-X45-SUP8L-E";
        range 192.0.2.10 192.0.2.50;
    }
    pool {
        allow members of "WS-4500X-16";
        range 192.0.2.51 192.0.2.100;
    }
}

subnet 203.0.113.0 netmask 255.255.255.0 {
    range 203.0.113.12 203.0.113.100;
    option routers 198.51.100.3;
    option subnet-mask 255.255.255.0;
    server-identifier 198.51.100.2;
    next-server 198.51.100.2;
}

Using TFTP

allow booting;
allow bootp;
ddns-update-style none;
#DEFINE AN OPTION SPACE. "EXAMPLE" IS USED HERE. IT IS A VARIABLE YOU CAN SET.
#MAINTAIN code 1,2 AND 3 CONSISTENTLY SINCE THE VALUES CORRESPOND TO CONFIG,SCRIPT AND
#OVA FILES RESPECTIVELY.
option space EXAMPLE;
option EXAMPLE.startup-config code 1=text;
option EXAMPLE.user-script code 2=text;
option EXAMPLE.user-ova code 3=text;

#ENTER THESE DETAILS AS APPLICABLE TO YOUR NETWORK.
option domain-name "example.com";
option domain-name-servers 198.51.100.3;
option subnet-mask 255.255.255.0;
option broadcast-address 192.0.2.255;

#DEFINE A CLASS FOR THE VENDOR-SPECIFIC IDENTIFIER NAME THAT THE DEVICE HAS.
#EXAMPLE: FOR SUP8E/8LE IT IS "WS-X45-SUP8L-E"
#FOR CATALYST 4500-X IT IS "WS-4500X-16"
#ALSO DEFINE THE ROUTER,TFTP SERVER IDENTIFIER,NEXT SERVER IP DETAILS - AS APPLICABLE
#TO YOUR NETWORK
```
class "WS-X45-SUP8L-E" {
    match pick-first-value (option dhcp-client-identifier, hardware);
    #THE OPTION ROUTER ADDRESS IS REQUIRED ONLY IF YOU USE A RELAY AGENT BETWEEN THE
    #DHCP SERVER AND THE CLIENT.
    option routers 198.0.2.254;
    option subnet-mask 255.255.255.0;
    server-identifier 198.51.100.2;
    next-server 198.51.100.2;
}

class "WS-4500X-16" {
    match pick-first-value (option dhcp-client-identifier, hardware);
    option routers 198.0.2.254;
    option subnet-mask 255.255.255.0;
    server-identifier 198.51.100.2;
    next-server 198.51.100.2;
}

#DEFINE A SUBCLASS TO ADD THE DEVICE BASED ON IT'S MAC ADDRESS TO RECEIVE
#CONFIGURATION FILES. THIS APPLIES WHEN YOU HAVE MULTIPLE DEVICES WITH SAME
#VENDOR-SPECIFIC IDENTIFIER AND YOU WANT TO PUSH DIFFERENT CONFIGURATIONS BASED ON THE
#MAC ADDRESS

    # ENTER THE BOOTFILENAME.THE .BIN IMAGE FILE SHOULD RESIDE IN THE TFTP BOOT FOLDER.
    filename "example2-ios-image.bin";
    option routers 198.0.2.254;

    #SPECIFY THAT THE OPTION 43 AND ROUTER(3) DETAILS HAVE TO BE SENT TO THE CLIENT SWITCH
    option dhcp-parameter-request-list 43,3;
    option vendor-class-identifier "WS-X45-SUP8L-E";
    vendor-option-space EXAMPLE;

    #SPECIFY THE PATH OF THE FILES YOU WANT TO SEND (TFTP).
    #MAKE SURE THESE FILES RESIDE IN IDENTICAL FOLDERS (configs/,scripts/,container/) IN
    #the TFTP BOOT FOLDER. YOU MUST CREATE THE IDENTICAL FOLDERS WITH THE SAME NAME AND
    #CASE.
    #ENTER A FILE NAME. MAKE SURE THAT CONFIG, SCRIPT, AND OVA FILE EXTENTIONS ARE
    #<config-file>.config,<script-file>.script,<container-file>.ova RESPECTIVELY.

    option EXAMPLE.startup-config
    "tftp://198.51.100.2/configs/example2-config.config";
    option EXAMPLE.user-script "tftp://198.51.100.2/scripts/example2-script.py";
    option EXAMPLE.user-ova
    "tftp://198.51.100.2/container/example2_dmi_container.ova";
    option dhcp-parameter-request-list 43,3;
}

subclass "WS-X45-SUP8L-E" 1:e4:aa:5d:c4:a5:a1 {
    #WHEN USING TFTP TO DOWNLOAD FILES, PROVIDE THE PATH IN THE FOLLOWING FORMAT:
    #filename "tftp://<next-server ip address>/<ios_image.bin>";
    filename "tftp://198.51.100.2/example2-ios-image.bin"

    option routers 198.0.2.254;
    option dhcp-parameter-request-list 43,3;
    option vendor-class-identifier "WS-X45-SUP8L-E";
    vendor-option-space EXAMPLE;

    option EXAMPLE.startup-config
    "tftp://198.51.100.2/configs/example2-config.config";
    option EXAMPLE.user-script "tftp://198.51.100.2/scripts/example2-script.py";
    option EXAMPLE.user-ova
    "tftp://198.51.100.2/container/example2_dmi_container.ova";
option dhcp-parameter-request-list 43,3;
}

subclass "WS-4500X-16" 1:30:e4:db:f8:a4:9f {
  filename "tftp://198.51.100.2/example2-ios-image.bin";
  option routers 198.0.2.254;
  option dhcp-parameter-request-list 43,3;
  option vendor-class-identifier "WS-4500X-16";
  vendor-option-space EXAMPLE;
  option EXAMPLE.startup-config "tftp://198.51.100.2/configs/example2-config.config";
  option EXAMPLE.user-script "tftp://198.51.100.2/scripts/example2-script.py";
  option EXAMPLE.user-ova "tftp://198.51.100.2/container/example2_dmi_container.ova";
  option dhcp-parameter-request-list 43,3;
}

#ASSIGN A POOL TO GIVE IP ADDRESSES TO THE MEMBERS OF THE VENDOR-SPECIFIC CLASS
subnet 192.0.2.0 netmask 255.255.255.0 {
  pool {
    allow members of "WS-X45-SUP8L-E";
    range 192.0.2.10 192.0.2.50;
  }
  pool {
    allow members of "WS-4500X-16";
    range 192.0.2.51 192.0.2.100;
  }
}

subnet 203.0.113.0 netmask 255.255.255.0 {
  range 203.0.113.12 203.0.113.100;
  option routers 198.51.100.3;
  option subnet-mask 255.255.255.0;
  server-identifier 198.51.100.2;
  next-server 198.51.100.2;
}

Configuring DHCP Option 43 (for Microsoft Windows)

DHCP Option 43 is used by clients and servers to exchange vendor-specific information. (RFC 2132).

This section describes the DHCP Option 43 configuration information that pertains to sending device configuration files, script files, and .ova files to the switch. It is applicable only if you use OpenDhcpServer as the DHCP server, with Microsoft Windows. Other DHCP servers have their own methods to configure this option and the information is available on the Internet.

To send any file, you must convert the file name along with the extension, to a hexadecimal format and the files must be stored in the TFTP root directory.

<File code><length of filename.ext in hexadecimal value><hex value of the filename.ext>

Use the relevant codes to specify the type of file you want to send:

- code 01—A configuration file. For example, to send file example-config.config:

- code 02—A script file. For example, to send file example-script.py:

- code 03—An ova file. For example, to send file example_dmi_container.ova:

Concatenating all three file names
Microsoft Windows DHCP Server Configuration

The following example shows how to configure the DHCP Server on Microsoft Windows.

The example uses OpenDhcpServer and Solarwinds TFTP server. Information about configuring both is available on the Internet. The use of both applications here is only meant to serve as an example for configuration, and are not product recommendations.

Figure 6-3 Solarwinds TFTP Server

The important sections of this sample configuration are highlighted bold:

#This is a configuration file.
#Lines starting with punctuation marks are comments.
#This file should be saved in the same folder as the exe file.
#Remove; and replace the sample value with your own to change a setting

[LISTEN_ON]
#Specify the interfaces that the server should listen to. If you have more than one NIC
#card on your server, always specify the cards that will listen to DHCP/DNS requests.
#Requests from different interfaces look for matching DHCP ranges.
#Requests from relay agents look for a matching range to relay the agent IP.
#You can specify up to 125 interfaces
#By default this includes all static interfaces
;198.51.100.2

[LOGGING]
#You can set the LogLevel as None, Errors or All
#We recommend that you set the logging level to Normal, Normal include errors and DHCP
#renewal messages. The default logging level is Normal.
;LogLevel=None
;LogLevel=Normal
;LogLevel=All
;LogLevel=Debug

[REPLICATION_SERVERS]
#You can have 2 instances of Open DHCP Servers in a network. Open DHCP Server sends
#replicated inform messages to the other instance of the Open DHCP server and replicates
#leases. The IP address allotted by one server is not reallocated by the other server to
#another host. Further when one server goes down, the other can renew the leases, without
#NAR and DISCOVER. You must specify the primary and secondary servers for replication to
#work. Ensure that the primary & secondary server entries are identical on both servers.
#You may copy the entire ini file on both servers and change the
#LISTEN_ON on individual servers, if needed.
;Primary=192.0.2.253
;Secondary=192.0.2.254

[HTTP_INTERFACE]
#This is http interface to display the lease status. By default, this is the fist
#interface, port 6789. You can change it to any network interface.
;HTTPServer=192.51.100.1:6789
#To limit client access, you can specify up to 8 HTTP client IPs here. If no client IPs
#are specified then clients can access the HTTP interface.
#You can also change the title of the HTML page
;HTTPTitle=example-httpserver-url.com

#Sections below are other DHCP Sections. You can allot clients addresses dynamically from
#the DHCP Range or statically. For static addresses, client section must be created for
#each static client against its MAC Address. BOOTP clients are always static. DHCP Ranges
#are grouped into [RANGE_SET]s, so that range specific options can be specified for a
#group of ranges in one place. The total ranges in all [RANGE_SET]s is also 125 and you
#can also have a maximum of 125 [RANGE_SET]s.
#You can specify one or more ranges in each [RANGE_SET] section, in the specified format.
#Open DHCP Server allots addresses from these ranges. Static Hosts and BootP clients do
#not require ranges. You do not have to specify a [RANGE_SET] or a DHCP_Range if all
#clients are Static.
#The dynamic address allocation policy is -
#1)Look to see if a MAC address is specified as static DHCP Client and use that IP
#2)If not found, look for an old, expired or active address of the same host
#3)If not found, look at the requested IP address and check if available
#4)If not found, allot an unused IP address, if available
#5)If not found, allot the expired IP address of the other host.
#From 2) to 6), requests from different interfaces look for matching DHCP ranges
#of interface IP, and requests from relay agents look for matching range to
#relay agent IP.

#All the ranges in a [RANGE_SET] section can be further restricted with Filter_Mac_Range,
#Filter_Vendor_Class and Filter_User_Class. For example, if a MAC range is specified, then
#this section's ranges will only be available to hosts, who's MAC address falls in this
#range. Also if any host has a matching Filter_Mac_Range in any DHCP_RANGE section, then
#other DHCP range sections without a Filter_Mac_Range or without a matching MAC range will
#not be available to it. Each Manufacturer has a fixed MAC Range. The same Mac ranges can
#repeat in many DHCP_RANGE sections.
#For Filter_Vendor_Class (option 60) and Filter_User_Class filter (option 77),
#the range is available only to a matching value of Filter_Vender_Class
#and Filter_User_Class sent in the client request. If the Filter_Vender_Class and
#the Filter_User_Class values do match in one or more ranges, other ranges with missing
#or non-matching values are not available to such clients. You can specify upto 32
#Filter_Mac_Range, Filter_Vender_Class and Filter_User_Class in each [RANGE_SET].
Generally, you do not have to specify any filters for the relay agent. The range is automatically selected based on the relay agent IP and subnet mask of the range. The relay agent always sends its subnet side IP. This server uses only the DHCP Range, which matches this IP. This ensures that correct range is used. This feature eliminates the need for additional configuration. For matching purpose, range is recalculated using Subnet Mask of range and Relay Agent IP. However, if you want to manually configure the subnet selection, you can use FilterSubnetSelection in a RANGE_SET. If this filter is specified, it will be first matched with SubnetSelection Option 118 sent by client. If client sends no such option, it will be matched with relay agent IP. If not relay agent IP is sent, Listening Interface’s IP will be matched. You can also override the Target Relay Agent using TargetRelayAgent Option.

[RANGE_SET]
# This is first and simple DHCP range section example,
# if you need range filters, look at example below
DHCPRange=192.0.2.3-192.0.2.250
VendorClass="Example Server"
# The following are the range specific DHCP options.
# You can copy more options names from [GLOBAL_OPTIONS]
SubnetMask=255.255.255.0
;DomainServer=198.51.100.3
Router=198.0.2.254
# Lease Time can be different for this Range
;AddressTime=360

[RANGE_SET]
# This section is also simple [RANGE_SET] section
# Here the options are specified as flat options.
;DHCPRange=192.0.2.3-192.0.2.250
# The following are the flat range specific DHCP options.
# SubnetMask below
;1=255.255.255.0
# DomainServers below
;6=198.51.100.3
# Router
;3=198.0.2.254
# AddressTime
;51=11000

[RANGE_SET]
# This is filtered [RANGE_SET] section.
# The first eight entries in this example are filters.
# Currently, only the following types of filters are supported.
# However, 32 filters of each type can be specified
;FilterMacRange=00:0d:60:c5:4e:00-00:0d:60:c5:4e:ff
;FilterMacRange=00:0e:12:c5:4e:00-00:0e:12:c5:4e:ff
;FilterMacRange=00:0f:60:c5:4e:a1-00:0f:60:c5:4e:a1
;FilterVendorClass="EXAMPLE 5.0"
;FilterVendorClass="EXAMPLE 5.1"
;FilterVendorClass="EXAMPLE 5.2"
;FilterUserClass="My User Class 4.0"
;FilterUserClass=123,56,87,123,109,0,23,56,156,209,234,56
;FilterUserClass=00:0d:60:c5:4e:0d:60:c5:4e
# You can select RANGE_SET based on FilterSubnetSelection
;FilterSubnetSelection=198.51.100.1
;FilterSubnetSelection=192.0.2.1
;Ethernet=no

[GLOBAL_OPTIONS]
#These are global DHCP Options and they supplement client specific options and [RANGE_SET] options. Options tags start with 1 and go up to 254, you can specify and option like #1=255.255.255.0, but it may be difficult to remember option tags. Try using Option names instead. If a matching name is not found, you can use tag=value (flat options)
#You can also specify the value as byte array or even hex array. Some options that have #sub-options can be specified only as hex/byte array. If options have client specific #values, move/copy them to specific static client sections. If any option has a DHCP range #specific value, move or copy them to [RANGE_SET] sections.
#You may quote string values (must quote if string contain chars like comma, dot or #colon). For example NDS_Tree_Name="my.NDS.Tree"or 43="this is return string" or use the #byte array in value like 43=123,56,87,123,109,0,23,56,156,209,234,56 or use the hex #array in value 43=a6:b6:c9:ae:eb:89:09:a4:67:d5

;DomainName="example-httpserver-url.com"
;SubNetMask=255.255.255.0
;DomainServer=198.51.100.3
;Router=192.0.2.254
#AddressTime is default lease time for server
#specify 0 for infinity lease time
<AddressTime=36000
;RenewalTime=0
;RebindingTime=0
#NextServer is PXEBoot TFTP Server
NextServer=198.51.100.2
;Trainers=yes
;ARPTimeout=3453
;Ethernet=yes
;DefaultTCPTTL=21
;KeepaliveTime=120
;KeepaliveData=yes
;TFTPServerName=MyTFTPServer
BootFileName=example-ios-image.bin
;AutoConfig=yes
;NameServiceSearch=23,0,235,4,2,0,236,7,94,34,87,4,127,254,23
;SubnetSelectionOption=255.255.255.240
#Option TFTPServerIPaddress is for phone use only, for PXEBoot use NextServer option
;/TFTPServerIPaddress=198.51.100.2

#Following sections are Static Client DHCP entries/options
#if no IP is given, then that host will never be allotted any IP
#More option Names can be copied from DHCP-OPTIONS to clients.
#for BOOTP requests, only these options would be sent.
#For DHCP requests. Missing Options will be supplemented from #first [DHCP-RANGE] options (if IP falls in any range), other #options will be supplemented from [DHCP-OPTIONS].

[00:41:42:41:42:00]
#This is a client with MAC addr 00:41:42:41:42:00
IP=192.0.2.200
#No other options specified for this client
#For non BOPT requests, Missing Options will be supplemented from first [RANGE_SET] #options, if IP falls in any range. and other missing would be added from [GLOBAL_OPTIONS].

[00:41:42:41:42:05]
#This is a client with MAC addr 00:41:42:41:42:05
IP=192.0.2.201
#DHCP will offer following hostname to this client
;HostName=TestHost
[00:ff:a4:0e:ef:99]
#This host has no IP
#This host will not get an IP, even from Dynamic Ranges
#You can use such entries to prevent a host from getting an IP from this Server.

Autoboot Process Output

Autoboot Process Output—Using HTTP, page 6-37
Autoboot Process Output—Using TFTP, page 6-40

Autoboot Process Output—Using HTTP

The following is sample output of the autoboot process on Cisco Catalyst 4500-X Series Switches.

- The HTTP server from which the files are being downloaded is example-httpserver-url.com.
- The image, configuration, script, and ova files being downloaded are example-ios-image.bin, example-config.config, example-script.py, and example_dmi_container.ova respectively.

rommon 1 >reset
Resetting .......

rommon 2 >
Rommon (G) Signature verification PASSED
Rommon (P) Signature verification PASSED
FPGA   (P) Signature verification PASSED

***************************************************************
*                                                            *
* Welcome to Rom Monitor for WS-C4500X-16 System.            *
* Copyright (c) 2008-2014 by Cisco Systems, Inc.             *
* All rights reserved.                                       *
*                                                            *
***************************************************************

Rom Monitor (P) Version 15.0(1r)SG14
CPU Rev: 2.2, Board Rev: 3, Board Type: 108
CPLD Mobat Rev: 2.0x4377.0xb277
Chassis: WS-C4500X-16

MAC Address : 30-e4-db-f8-a4-7f
Ip Address   : Not set.
Netmask      : Not set.
Gateway      : Not set.
TftpServer   : Not set.

Non-Redundant system or peer not running IOS
System Uplinks & Linecards have been reset!!

***** The system will autoboot in 5 seconds *****

Type control-C to prevent autoboooting.
...
Management Ethernet Link Up: 1Gb Full Duplex
Received DHCP_ACK.
Extending autoboot timeout ...

DHCP Bootfile:http://example-httpserver-url.com/example-ios-image.bin

HTTP Session Details are ...

Filename : /example-ios-image.bin
IP Address : 192.0.2.1
HttpServer : 198.51.100.1

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!

Loaded 197495364 bytes successfully.

Checking digital signature....
[http://example-httpserver-url.com/example-ios-image.bin]
Digitally Signed Development Software with key version A

Rommon reg: 0x00084F80
Reset2Reg: 0x00004F00

Image load status: 0x00000000
###
Winter 110 controller 0x0468AFAC..0x047F4313 Size:0x002FDB9D

Program Done!

[0.091269] pci 0000:00:00.0: ignoring class b20 (doesn't match header type 01)
[0.181505] pci 0001:04:00.0: ignoring class b20 (doesn't match header type 01)
[0.274669] pci 0002:0c:00.0: ignoring class b20 (doesn't match header type 01)

Starting System Services

devpts /dev/pts devpts rw, nosuid, noexec, relatime, gid=4, mode=600, ptxmode=000 0 0

diagsk10-post version 5.1.4.1

prod: WS-C4500X-16 part: 73-13860-03 serial: JAE155209ZD

Power-on-self-test for Module 1: WS-C4500X-16

CPU Subsystem Tests ... 
seeprom: Pass

Traffic: L3 Loopback ... 
Test Results: Pass

Traffic: L2 Loopback ... 
Test Results: Pass
post done(57 secs)
Exiting to ios...

Downloading http://example-httpserver-url.com/example-config.config

<table>
<thead>
<tr>
<th>Total</th>
<th>Received</th>
<th>Xferd</th>
<th>Average Speed</th>
<th>Time</th>
<th>Time</th>
<th>Time</th>
<th>Time</th>
<th>Current</th>
<th>Dload</th>
<th>Upload</th>
<th>Total</th>
<th>Spent</th>
<th>Left</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2267</td>
<td>100</td>
<td>2267</td>
<td>0</td>
<td>0</td>
<td>222k</td>
<td>0</td>
<td>--:--:--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1106k</td>
</tr>
</tbody>
</table>

Downloading http://example-httpserver-url.com/example-script.py

<table>
<thead>
<tr>
<th>Total</th>
<th>Received</th>
<th>Xferd</th>
<th>Average Speed</th>
<th>Time</th>
<th>Time</th>
<th>Time</th>
<th>Time</th>
<th>Current</th>
<th>Dload</th>
<th>Upload</th>
<th>Total</th>
<th>Spent</th>
<th>Left</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2391</td>
<td>100</td>
<td>2391</td>
<td>0</td>
<td>0</td>
<td>712k</td>
<td>0</td>
<td>--:--:--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2334k</td>
</tr>
</tbody>
</table>
Continuing with IOS boot...

Nov 21 09:06:06 %IOSXE-3-PLATFORM: process kernel: [ 93.350890] mpc85xx_pci_err_probe: Unable to request irq 0 for MPC85xx PCI err
Nov 21 09:06:06 %IOSXE-3-PLATFORM: process kernel: [ 93.361062] mpc85xx_pcie_err_probe: Unable to request irq 0 for MPC85xx PCIe err

Loading gsbu64atomic as gdb64atomic
Loading pds_kc_flowcntl for kstack
loading kstack module
Loading container module
Failed to bring interface "eth1" up
Using 1 for MTS slot
Platform Manager: starting in standalone mode (active)

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR sec. 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS sec. 252.227-7013.

cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3 Switch Software (cat4500e-UNIVERSALK9-M), Version 03.09.01.E.179 EARLY DEPLOYMENT [PROD IMAGE] ENGINEERING NOVA WEEKLY BUILD, synced to V152.5.1.68_E1
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Sat 12-Nov-16 19:26 by sdcunha

Cisco IOS-XE software, Copyright (c) 2005-2015 by cisco Systems, Inc. All rights reserved. Certain components of Cisco IOS-XE software are licensed under the GNU General Public License ("GPL") Version 2.0. The software code licensed under GPL Version 2.0 is free software that comes with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such GPL code under the terms of GPL Version 2.0. (http://www.gnu.org/licenses/gpl-2.0.html) For more details, see the documentation or "License Notice" file accompanying the IOS-XE software, or the applicable URL provided on the flyer accompanying the IOS-XE software.

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at: http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to export@cisco.com.
Autoboot Process Output—Using TFTP

The following is sample output of the autoboot process on Cisco Catalyst 4500E Series Switches with Supervisor Engine 8-E.

- The TFTP server from which the files are being downloaded is 198.51.100.2.
- The image, configuration, script and, ova files being downloaded are example2-ios-image.bin, example2-config.config, example2-script.py, and example2_dmi_container.ova respectively.

```
rommon 1 >reset
Reseting .......
Verifying FPGA (P) Signature ....................... PASSED
Verifying ROMMON (P) Signature ........... PASSED

Rom Monitor (P) Version 15.1(1r)SG8
Compiled Wed 26-Oct-16 12:13 [RLS]
Chassis      : WS-C4507R*E    Mod  [1]
Revision     : CPU 2.1   BOARD 3.0   FPGA 4.3571.7DC7
Memory       : 4096 MB
Date         : Mon Nov 21 09:14:09 2016

***** The system will autoboot in 5 seconds *****
```

Type Control-C to prevent autobooting....
Sending DHCP_DISCOVER Link Speed   : 1Gb Full Duplex
Received DHCP_ACK .
DHCP state: DHCP_BOUND

DHCP Bootfile: tftp://198.51.100.2/example2-ios-image.bin
Link Speed   : 1Gb Full Duplex
Filename     : /example2-ios-image.bin
IpAddress    : 192.0.2.2
TftpServer   : 198.51.100.2
TftpBlkSize  : 1468
Checking digital signature....
Digitally Signed Development Software with key version A

Rommon reg: 0x00084F80
Reset2Reg: 0x0CB00000

ConanLite controller 0x381D7988..0x38488CC0 Size: 0x0074D07C @

Radtrooper controller 0x37AEB588..0x37C87122 Size: 0x00661EDC @

Program Done!

Checking digital signature....
Digitally Signed Development Software with key version A

pci 0000:00:00.0: ignoring class b20 (doesn't match header type 01)
pci 0001:02:00.0: ignoring class b20 (doesn't match header type 01)
pci 0002:04:00.0: ignoring class b20 (doesn't match header type 01)
audit: cannot initialize inotify handle
All packages are Digitally Signed
Starting System Services
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=4,mode=600,ptmxmode=000 0 0
diagsk10-post version 6.2.0.0

prod: WS-X45-SUP8L-E part: 73-16780-03 serial: CAT1940L26Y

Power-on-self-test for Module 3: WS-X45-SUP8L-E

CPU Subsystem Tests ...
seeprom: Pass
Traffic: L3 Loopback ...
Test Results: Pass
Traffic: L2 Loopback ...
Test Results: Pass
post done(64 secs)
Exiting to ios...

Downloading tftp://198.51.100.2/configs/example2-config.config
% Total  % Received  % Xferd Average Speed Time Time Current
   Dload  Upload   Total    Spent    Left  Speed
100  5848  100  5848  0  0  441k  0  --:--:-- --:--:-- --:--:--  441k

Downloading tftp://198.51.100.2/scripts/example2-script.py
% Total  % Received  % Xferd Average Speed Time Time Current
   Dload  Upload   Total    Spent    Left  Speed
100  2391  100  2391  0  0  603k  0  --:--:-- --:--:-- --:--:--  603k
Chapter 6      Programmability

Sample Configuration and Reference Information

Downloading tftp://198.51.100.2/container/example2_dmi_container.ova

Continuing with IOS boot..
Loading gsbu64atomic as gdb64atomic
Loading pds_kc_flowcnt1 for kstack
Loading kstack module
Loading container module
Using 3 for MTS slot
Platform Manager: starting in standalone mode (active)

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR sec. 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS sec. 252.227-7013.

cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3 Switch Software (cat4500esb-UNIVERSALK9-M), Version 03.09.01.E.179 EARLY DEPLOYMENT [PROD IMAGE] ENGINEERING NOVA WEEKLY BUILD, synced to V152_5_1_68_E1 Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc. Compiled Sat 12-Nov-16 13:49 by sdcunha

Cisco IOS-XE software, Copyright (c) 2005-2015 by cisco Systems, Inc. All rights reserved. Certain components of Cisco IOS-XE software are licensed under the GNU General Public License ("GPL") Version 2.0. The software code licensed under GPL Version 2.0 is free software that comes with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such GPL code under the terms of GPL Version 2.0. (http://www.gnu.org/licenses/gpl-2.0.html) For more details, see the documentation or "License Notice" file accompanying the IOS-XE software, or the applicable URL provided on the flyer accompanying the IOS-XE software.

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at: http://www.cisco.com/wwl/export/crypto/tool/stqrgh.html

If you require further assistance please contact us by sending email to export@cisco.com.

cisco WS-C4507R+E (P5040) processor (revision 2) with 4194304K bytes of physical memory.
Processor board ID FXS1851Q140
P5040 CPU at 1.8Ghz. Supervisor 8L-E
Last reset from Reload

Cisco Catalyst 4500 Series Switches, Cisco IOS Software Configuration Guide - Cisco IOS XE 3.10.xE
1 Virtual Ethernet interface
48 Gigabit Ethernet interfaces
8 Ten Gigabit Ethernet interfaces
511K bytes of non-volatile configuration memory.

**********************************************************
* WARNING WARNING WARNING !!!!!!!!                       *
*                                                        *
* The ISSU compatibility matrix check has been disabled. *
* No image version compatibility checking will be done.  *
* Please be sure this is your intention.                 *
**********************************************************

Press RETURN to get started!

Switch>