



Performing Device Setup Configuration

- [Restrictions for Performing Device Setup Configuration, on page 1](#)
- [Information About Performing Device Setup Configuration, on page 1](#)
- [Device Boot Process, on page 2](#)
- [Software Install Overview, on page 3](#)
- [Software Boot Modes, on page 3](#)
- [Installed Boot Mode, on page 3](#)
- [Bundle Boot Mode, on page 6](#)
- [Changing the Boot Mode, on page 6](#)
- [Installing the Software Package, on page 7](#)
- [Terminating a Software Install, on page 7](#)
- [Devices Information Assignment, on page 8](#)
- [Default Switch Information, on page 8](#)
- [DHCP-Based Autoconfiguration Overview, on page 8](#)
- [DHCP-Based Autoconfiguration and Image Update, on page 10](#)
- [DHCP Server Configuration Guidelines, on page 11](#)
- [How to Obtain Configuration Files, on page 12](#)
- [Scheduled Reload of the Software Image, on page 13](#)
- [How to Control Environment Variables, on page 14](#)
- [How to Perform Device Setup Configuration, on page 16](#)
- [Configuration Examples for Device Setup Configuration, on page 26](#)

Restrictions for Performing Device Setup Configuration

- Subpackage software installation is not supported.

Information About Performing Device Setup Configuration

The following sections provide information about how to perform a device setup configuration, including IP address assignments and Dynamic Host Configuration Protocol (DHCP) auto configuration.

Device Boot Process

The normal boot process involves the operation of the boot loader software and includes these activities:

- Performs low-level CPU initialization. This process initializes the CPU registers that control where physical memory is mapped, the quantity and speed of the physical memory, and so forth.
- Initializes the file systems on the system board.
- Loads a default operating system software image into memory and boots up the device.
- Performs power-on self-test (POST) for the CPU subsystem and tests the system DRAM.

The boot loader provides access to the file systems before the operating system is loaded. Normally, the boot loader is used only to load, decompress, and start the operating system. After the boot loader gives the operating system control of the CPU, the boot loader is not active until the next system reset or power-on.

Before you can assign device information, make sure you have connected a PC or terminal to the console port or a PC to the Ethernet management port, and make sure you have configured the PC or terminal-emulation software baud rate and character format to match these of the device console port:

- Baud rate default is 9600.
- Data bits default is 8.



Note If the data bits option is set to 8, set the parity option to none.

- Stop bits default is 2 (minor).
- Parity settings default is none.

Power-on Self Test

Power-on self test (POST) is a series of tests as part of its power-up procedure. The power-on self test (POST) diagnostic program performs these tests:

- checksum tests on the boot firmware in PROM and the switch firmware in Flash memory,
- internal data loopback test on all ports, and
- access and integrity tests on the ASIC.

POST runs during IOS boot and performs these tests:

- PHY and MAC loopback test to verify the data path between the CPU and network ports, and
- thermal test to verify the temperature reading from the device sensor.



Note You can disable POST using **boot skip-post** command in configuration mode in In Cisco ESS-9300-8X16T of the Curtiss-Wright VPX3-623.

For more information on the POST, see [POST](#)

Software Install Overview

The Software Install feature provides a uniform experience across different types of upgrades, such as full image install and Software Maintenance Upgrade (SMU)

The Software Install feature facilitates moving from one version of the software to another version in install mode. Use the **install** command in privileged EXEC mode to install or upgrade a software image. You can also downgrade to a previous version of the software image, using the install mode.

The method that you use to upgrade Cisco IOS XE software depends on whether the switch is running in install mode or in bundle mode. In bundle mode or consolidated boot mode, a .bin image file is used from a local or remote location to boot the device. In the install boot mode, the bootloader uses the packages.conf file to boot up the device.

The following software install features are supported on your switch:

- Software bundle installation.
- Software rollback to a previously installed package set.
- Emergency installation in the event that no valid installed packages reside on the boot flash.

Software Boot Modes

Your device supports two modes to boot the software packages:

- Installed mode
- Bundle mode

Installed Boot Mode

You can boot your device in installed mode by booting the software package provisioning file that resides in flash:

```
Switch: boot flash:packages.conf
```



Note The packages.conf file for particular release is created on following the install workflow described in the section, *Installing a Software Package*.

The provisioning file contains a list of software packages to boot, mount, and run. The ISO file system in each installed package is mounted to the root file system directly from flash.



Note The packages and provisioning file used to boot in installed mode must reside in flash. Booting in installed mode from usbflash0: or tftp: is not supported.

Installing a Software Package

You can install, activate, and commit a software package using a single command or using separate commands. This task shows how to use the **install add file activate commit** command for installing a software package.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	install add file tftp: filename [activate commit] Example:	Copies the software install package from a remote location (via FTP, HTTP, HTTPS, TFTP) to the device, performs a compatibility check for the platform and image versions, activates the software package, and makes the package persistent across reloads. <ul style="list-style-type: none"> • This command extracts the individual components of the .bin file into sub-packages and packages.conf file. • The device reloads after executing this command.
Step 3	exit Example: Device# exit	Exits privileged EXEC mode and returns to user EXEC mode.

Managing the Update Package

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	install add file tftp: <i>filename</i> Example: <pre>Device# install add file tftp://172.16.0.1/tftpboot/folder1/ ess9300_iosxe.17.04.01.SPA.bin</pre>	<p>Copies the software install package from a remote location (via FTP, HTTP, HTTPS, TFTP) to the device, and performs a compatibility check for the platform and image versions.</p> <ul style="list-style-type: none"> This command extracts the individual components of the .bin file into sub-packages and packages.conf file.
Step 3	install activate [auto-abort-timer] Example: <pre>Device# install activate</pre>	<p>Activates the added software install package, and reloads the device.</p> <ul style="list-style-type: none"> When doing a full software install, do not provide a package filename. The auto-abort-timer keyword, automatically rolls back the software image activation. <p>The automatic timer is triggered after the new image is activated. If the timer expires prior to the issuing of the install commit command, then the install process is automatically terminated. The device reloads, and boots up with a previous version of the software image.</p>
Step 4	install abort Example: <pre>Device# install abort</pre>	<p>(Optional) Terminates the software install activation, and rolls back to the version that was running before current installation procedure.</p> <ul style="list-style-type: none"> You can use this command only when the image is in an activated state; and not when the image is in a committed state.
Step 5	install commit Example: <pre>Device# install commit</pre>	<p>Makes the changes persistent over reload.</p> <ul style="list-style-type: none"> The install commit command completes the new image installation. Changes are persistent across reloads until the auto-abort timer expires.
Step 6	install rollback to committed Example: <pre>Device# install rollback to committed</pre>	<p>(Optional) Rolls back the update to the last committed version.</p>
Step 7	install remove { file filesystem: <i>filename</i> inactive } Example: <pre>Device# install remove inactive</pre>	<p>(Optional) Deletes all unused and inactive software installation files.</p>
Step 8	show install summary Example: <pre>Device# show install summary</pre>	<p>Displays information about the active package.</p> <ul style="list-style-type: none"> The output of this command varies according to the install commands that are configured.

Bundle Boot Mode

You can boot your device in bundle boot mode by booting the bundle (.bin) file:

The provisioning file contained in a bundle is used to decide which packages to boot, mount, and run. Packages are extracted from the bundle and copied to RAM. The ISO file system in each package is mounted to the root file system.

Unlike install boot mode, additional memory that is equivalent to the size of the bundle is used when booting in bundle mode.

Unlike install boot mode, bundle boot mode is available from several locations:

- flash:
- usbflash0:
- tftp:

Booting a Device in Bundle Mode

There are several methods by which you can boot the device — either by copying the bin file from the TFTP server and then boot the device, or by booting the device straight from flash or USB flash using the commands **boot flash:<image.bin>** or **boot usbflash0:<image.bin>** .

The following procedure explains how to boot the device from the TFTP server in the bundle mode.

Procedure

	Command or Action	Purpose
Step 1	switch:BOOT =<source path of .bin file> Example: switch:	Sets the boot parameters.
Step 2	boot Example: switch:boot	Boots the device.
Step 3	show version	(Optional) Displays the version of the image installed.

Changing the Boot Mode

To change a device running in bundle boot mode to install mode, set the boot variable to flash:packages.conf, and execute the **install add file flash:ess9300.bin activate commit** command. After the command is executed, the device reboots in install boot mode.

Installing the Software Package

You can install the software package on a device by using the **install add**, **install activate**, and **install commit** commands in privileged EXEC mode.

The **install add** command copies the software package from a local or remote location to the device. The location can be FTP, HTTP, HTTPS, or TFTP. The command extracts individual components of the .bin file into sub-packages and packages.conf file. It also validates the file to ensure that the image file is specific to the platform.

For the **install activate** command to work, the package must be available in the device bootflash. When this command is configured, previously added packages from the .bin file get activated, and the system reloads.

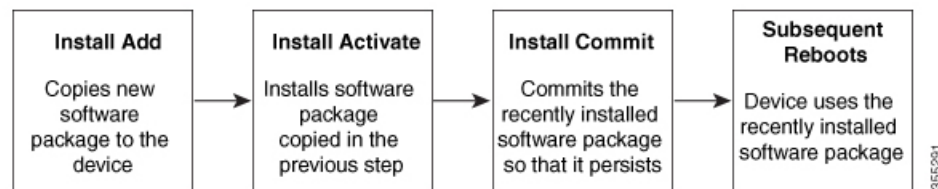
Enable the **install commit** command to make updates persistent over reloads.

Installing an update replaces any previously installed software image. At any time, only one image is installed on the device.

The following flow chart explains how the software install works:

Figure 1: Committing a Software Package

Process with Install Commit



Note The **install activate** command reloads the device with the new image.

Terminating a Software Install

You can terminate the activation of a software image in the following ways:

- Using the **install activate auto-abort-timer** command. When the device reloads after activating a new image, the auto-abort-timer is triggered. If the timer expires before issuing the **install commit** command, then the installation process is terminated; the device reloads again and boots up with the previous version of the software image.

Use the **install auto-abort-timer stop** command to stop this timer.

- Using the **install abort** command. This command rolls back to the version that was running before installing the new software. Use this command before issuing the **install commit** command.

Devices Information Assignment

You can assign IP information through the device setup program, through a DHCP server, or manually.

Use the device setup program if you want to be prompted for specific IP information. With this program, you can also configure a hostname and an enable secret password.

It gives you the option of assigning a Telnet password (to provide security during remote management) and configuring your switch as a command or member switch of a cluster or as a standalone switch.

Use a DHCP server for centralized control and automatic assignment of IP information after the server is configured.



Note If you are using DHCP, do not respond to any of the questions in the setup program until the device receives the dynamically assigned IP address and reads the configuration file.

If you are an experienced user familiar with the device configuration steps, manually configure the device. Otherwise, use the setup program described in section [Device Boot Process, on page 2](#).

Default Switch Information

Table 1: Default Switch Information

Feature	Default Setting
IP address and subnet mask	No IP address or subnet mask are defined.
Default gateway	No default gateway is defined.
Enable secret password	No password is defined.
Hostname	The factory-assigned default hostname is device.
Telnet password	No password is defined.
Cluster command switch functionality	Disabled.
Cluster name	No cluster name is defined.

DHCP-Based Autoconfiguration Overview

DHCP provides configuration information to Internet hosts and internetworking devices. This protocol consists of two components: one for delivering configuration parameters from a DHCP server to a device and an operation for allocating network addresses to devices. DHCP is built on a client-server model, in which designated DHCP servers allocate network addresses and deliver configuration parameters to dynamically configured devices. The device can act as both a DHCP client and a DHCP server.

During DHCP-based autoconfiguration, your device (DHCP client) is automatically configured at startup with IP address information and a configuration file.

With DHCP-based autoconfiguration, no DHCP client-side configuration is needed on your device. However, you need to configure the DHCP server for various lease options associated with IP addresses.

If you want to use DHCP to relay the configuration file location on the network, you might also need to configure a Trivial File Transfer Protocol (TFTP) server and a Domain Name System (DNS) server.

The DHCP server for your device can be on the same LAN or on a different LAN than the device. If the DHCP server is running on a different LAN, you should configure a DHCP relay device between your device and the DHCP server. A relay device forwards broadcast traffic between two directly connected LANs. A router does not forward broadcast packets, but it forwards packets based on the destination IP address in the received packet.

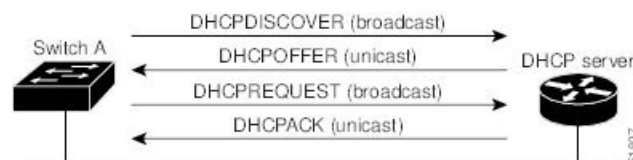
DHCP-based autoconfiguration replaces the BOOTP client functionality on your device.

DHCP Client Request Process

When you boot up your device, the DHCP client is invoked and requests configuration information from a DHCP server when the configuration file is not present on the device. If the configuration file is present and the configuration includes the **ip address dhcp** interface configuration command on specific routed interfaces, the DHCP client is invoked and requests the IP address information for those interfaces.

This is the sequence of messages that are exchanged between the DHCP client and the DHCP server.

Figure 2: DHCP Client and Server Message Exchange



The client, Device A, broadcasts a DHCPDISCOVER message to locate a DHCP server. The DHCP server offers configuration parameters (such as an IP address, subnet mask, gateway IP address, DNS IP address, a lease for the IP address, and so forth) to the client in a DHCPOFFER unicast message.

In a DHCPREQUEST broadcast message, the client returns a formal request for the offered configuration information to the DHCP server. The formal request is broadcast so that all other DHCP servers that received the DHCPDISCOVER broadcast message from the client can reclaim the IP addresses that they offered to the client.

The DHCP server confirms that the IP address has been allocated to the client by returning a DHCPACK unicast message to the client. With this message, the client and server are bound, and the client uses configuration information received from the server. The amount of information the device receives depends on how you configure the DHCP server.

If the configuration parameters sent to the client in the DHCPOFFER unicast message are invalid (a configuration error exists), the client returns a DHCPDECLINE broadcast message to the DHCP server.

The DHCP server sends the client a DHCPNAK denial broadcast message, which means that the offered configuration parameters have not been assigned, that an error has occurred during the negotiation of the parameters, or that the client has been slow in responding to the DHCPOFFER message (the DHCP server assigned the parameters to another client).

A DHCP client might receive offers from multiple DHCP or BOOTP servers and can accept any of the offers; however, the client usually accepts the first offer it receives. The offer from the DHCP server is not a guarantee that the IP address is allocated to the client; however, the server usually reserves the address until the client has had a chance to formally request the address. If the device accepts replies from a BOOTP server and configures itself, the device broadcasts, instead of unicasts, TFTP requests to obtain the device configuration file.

The DHCP hostname option allows a group of devices to obtain hostnames and a standard configuration from the central management DHCP server. A client (device) includes in its DHCPDISCOVER message an option 12 field used to request a hostname and other configuration parameters from the DHCP server. The configuration files on all clients are identical except for their DHCP-obtained hostnames.

DHCP-Based Autoconfiguration and Image Update

You can use the DHCP image upgrade features to configure a DHCP server to download both a new image and a new configuration file to one or more devices in a network. Simultaneous image and configuration upgrade for all switches in the network helps ensure that each new device added to a network receives the same image and configuration.

There are two types of DHCP image upgrades: DHCP autoconfiguration and DHCP auto-image update.

Restrictions for DHCP-Based Autoconfiguration

- The DHCP-based autoconfiguration with a saved configuration process stops if there is not at least one Layer 3 interface in an up state without an assigned IP address in the network.
- Unless you configure a timeout, the DHCP-based autoconfiguration with a saved configuration feature tries indefinitely to download an IP address.
- The auto-install process stops if a configuration file cannot be downloaded or if the configuration file is corrupted.
- The configuration file that is downloaded from TFTP is merged with the existing configuration in the running configuration but is not saved in the NVRAM unless you enter the **write memory** or **copy running-configuration startup-configuration** privileged EXEC command. If the downloaded configuration is saved to the startup configuration, the feature is not triggered during subsequent system restarts.

DHCP Autoconfiguration

DHCP autoconfiguration downloads a configuration file to one or more devices in your network from a DHCP server. The downloaded configuration file becomes the running configuration of the device. It does not overwrite the bootup configuration saved in the flash, until you reload the device.

DHCP Auto-Image Update

You can use DHCP auto-image upgrade with DHCP autoconfiguration to download both a configuration and a new image to one or more devices in your network. The devices (or devices) downloading the new configuration and the new image can be blank (or only have a default factory configuration loaded).

If the new configuration is downloaded to a switch that already has a configuration, the downloaded configuration is appended to the configuration file stored on the switch. (Any existing configuration is not overwritten by the downloaded one.)

To enable a DHCP auto-image update on the device, the TFTP server where the image and configuration files are located must be configured with the correct option 67 (the configuration filename), option 66 (the DHCP server hostname) option 150 (the TFTP server address), and option 125 (description of the Cisco IOS image file) settings.

After you install the device in your network, the auto-image update feature starts. The downloaded configuration file is saved in the running configuration of the device, and the new image is downloaded and installed on the device. When you reboot the device, the configuration is stored in the saved configuration on the device.

DHCP Server Configuration Guidelines

Follow these guidelines if you are configuring a device as a DHCP server:

- You should configure the DHCP server with reserved leases that are bound to each device by the device hardware address.
- If you want the device to receive IP address information, you must configure the DHCP server with these lease options:
 - IP address of the client (required)
 - Subnet mask of the client (required)
 - DNS server IP address (optional)
 - Router IP address (default gateway address to be used by the device) (required)
- If you want the device to receive the configuration file from a TFTP server, you must configure the DHCP server with these lease options:
 - TFTP server name (required)
 - Boot filename (the name of the configuration file that the client needs) (recommended)
 - Hostname (optional)
- Depending on the settings of the DHCP server, the device can receive IP address information, the configuration file, or both.
- If you do not configure the DHCP server with the lease options described previously, it replies to client requests with only those parameters that are configured. If the IP address and the subnet mask are not in the reply, the device is not configured. If the router IP address or the TFTP server name are not found, the device might send broadcast, instead of unicast, TFTP requests. Unavailability of other lease options does not affect autoconfiguration.
- The device can act as a DHCP server. By default, the Cisco IOS DHCP server and relay agent features are enabled on your device but are not configured. (These features are not operational.)

Purpose of the TFTP Server

Based on the DHCP server configuration, the device attempts to download one or more configuration files from the TFTP server. If you configured the DHCP server to respond to the device with all the options required for IP connectivity to the TFTP server, and if you configured the DHCP server with a TFTP server name, address, and configuration filename, the device attempts to download the specified configuration file from the specified TFTP server.

If you did not specify the configuration filename, the TFTP server, or if the configuration file could not be downloaded, the device attempts to download a configuration file by using various combinations of filenames and TFTP server addresses. The files include the specified configuration filename (if any) and these files: `network-config`, `cisconet.cfg`, `hostname.config`, or `hostname.cfg`, where *hostname* is the device's current hostname. The TFTP server addresses used include the specified TFTP server address (if any) and the broadcast address (255.255.255.255).

For the device to successfully download a configuration file, the TFTP server must contain one or more configuration files in its base directory. The files can include these files:

- The configuration file named in the DHCP reply (the actual device configuration file).
- The `network-config` or the `cisconet.cfg` file (known as the default configuration files).
- The `router-config` or the `ciscotr.cfg` file (These files contain commands common to all device. Normally, if the DHCP and TFTP servers are properly configured, these files are not accessed.)

If you specify the TFTP server name in the DHCP server-lease database, you must also configure the TFTP server name-to-IP-address mapping in the DNS-server database.

If the TFTP server to be used is on a different LAN from the device, or if it is to be accessed by the device through the broadcast address (which occurs if the DHCP server response does not contain all the required information described previously), a relay must be configured to forward the TFTP packets to the TFTP server. The preferred solution is to configure the DHCP server with all the required information.

Purpose of the DNS Server

The DHCP server uses the DNS server to resolve the TFTP server name to an IP address. You must configure the TFTP server name-to-IP address map on the DNS server. The TFTP server contains the configuration files for the device.

You can configure the IP addresses of the DNS servers in the lease database of the DHCP server from where the DHCP replies will retrieve them. You can enter up to two DNS server IP addresses in the lease database.

The DNS server can be on the same LAN or on a different LAN from the device. If it is on a different LAN, the device must be able to access it through a router.

How to Obtain Configuration Files

Depending on the availability of the IP address and the configuration filename in the DHCP reserved lease, the device obtains its configuration information in these ways:

- The IP address and the configuration filename is reserved for the device and provided in the DHCP reply (one-file read method).

The device receives its IP address, subnet mask, TFTP server address, and the configuration filename from the DHCP server. The device sends a unicast message to the TFTP server to retrieve the named configuration file from the base directory of the server and upon receipt, it completes its boot up process.

- The IP address and the configuration filename is reserved for the device, but the TFTP server address is not provided in the DHCP reply (one-file read method).

The device receives its IP address, subnet mask, and the configuration filename from the DHCP server. The device sends a broadcast message to a TFTP server to retrieve the named configuration file from the base directory of the server, and upon receipt, it completes its boot-up process.

- Only the IP address is reserved for the device and provided in the DHCP reply. The configuration filename is not provided (two-file read method).

The device receives its IP address, subnet mask, and the TFTP server address from the DHCP server. The device sends a unicast message to the TFTP server to retrieve the network-config or cisco.net.cfg default configuration file. (If the network-config file cannot be read, the device reads the cisco.net.cfg file.)

The default configuration file contains the hostnames-to-IP-address mapping for the device. The device fills its host table with the information in the file and obtains its hostname. If the hostname is not found in the file, the device uses the hostname in the DHCP reply. If the hostname is not specified in the DHCP reply, the device uses the default *Switch* as its hostname.

After obtaining its hostname from the default configuration file or the DHCP reply, the device reads the configuration file that has the same name as its hostname (*hostname-config* or *hostname.cfg*, depending on whether network-config or cisco.net.cfg was read earlier) from the TFTP server. If the cisco.net.cfg file is read, the filename of the host is truncated to eight characters.

If the device cannot read the network-config, cisco.net.cfg, or the hostname file, it reads the router-config file. If the device cannot read the router-config file, it reads the ciscortr.cfg file.



Note The device broadcasts TFTP server requests if the TFTP server is not obtained from the DHCP replies, if all attempts to read the configuration file through unicast transmissions fail, or if the TFTP server name cannot be resolved to an IP address.

Scheduled Reload of the Software Image

You can schedule a reload of the software image to occur on the device at a later time (for example, late at night or during the weekend when the device is used less), or you can synchronize a reload network-wide (for example, to perform a software upgrade on all device in the network).



Note A scheduled reload must take place within approximately 24 days.

You have these reload options:

- Reload of the software to take affect in the specified minutes or hours and minutes. The reload must take place within approximately 24 hours. You can specify the reason for the reload in a string up to 255 characters in length.

- Reload of the software to take place at the specified time (using a 24-hour clock). If you specify the month and day, the reload is scheduled to take place at the specified time and date. If you do not specify the month and day, the reload takes place at the specified time on the current day (if the specified time is later than the current time) or on the next day (if the specified time is earlier than the current time). Specifying 00:00 schedules the reload for midnight.

The **reload** command halts the system. If the system is not set to manually boot up, it reboots itself.

If your device is configured for manual booting, do not reload it from a virtual terminal. This restriction prevents the device from entering the boot loader mode and then taking it from the remote user's control.

If you modify your configuration file, the device prompts you to save the configuration before reloading. During the save operation, the system requests whether you want to proceed with the save if the `CONFIG_FILE` environment variable points to a startup configuration file that no longer exists. If you proceed in this situation, the system enters setup mode upon reload.

To cancel a previously scheduled reload, use the **reload cancel** privileged EXEC command.

How to Control Environment Variables

With a normally operating device, you enter the boot loader mode only through the console connection configured for 9600 bps.

The device boot loader software provides support for nonvolatile environment variables, which can be used to control how the boot loader, or any other software running on the system, operates. Boot loader environment variables are similar to environment variables that can be set on UNIX or DOS systems.

Environment variables that have values are stored in flash memory outside of the flash file system.

Each line in these files contains an environment variable name and an equal sign followed by the value of the variable. A variable has no value if it is not present; it has a value if it is listed even if the value is a null string. A variable that is set to a null string (for example, “”) is a variable with a value. Many environment variables are predefined and have default values.

You can change the settings of the environment variables by accessing the boot loader or by using Cisco IOS commands. Under normal circumstances, it is not necessary to alter the setting of the environment variables.

Common Environment Variables

This table describes the function of the most common environment variables.

Table 2: Common Environment Variables

Variable	Boot Loader Command	Cisco IOS Global Configuration Command
BOOT	<p>set BOOT <i>filesystem :/file-url ...</i></p> <p>A semicolon-separated list of executable files to try to load and execute when automatically booting.</p>	<p>boot system <i>{filesystem :/file-url ... switch {number all}}</i></p> <p>Specifies the Cisco IOS image to load during the next boot cycle. This command changes the setting of the BOOT environment variable.</p> <p>The package provisioning file, also referred to as the <i>packages.conf</i> file, is used by the system to determine which software packages to activate during boot up.</p> <ul style="list-style-type: none"> • When booting in installed mode, the package provisioning file specified in the boot command is used to determine which packages to activate. For example boot flash:packages.conf. • When booting in bundle mode, the package provisioning file contained in the booted bundle is used to activate the packages included in the bundle. For example, boot flash:image.bin.
MANUAL_BOOT	<p>set MANUAL_BOOT yes</p> <p>Decides whether the switch automatically or manually boots.</p> <p>Valid values are 1, yes, 0, and no. If it is set to no or 0, the boot loader attempts to automatically boot up the system. If it is set to anything else, you must manually boot up the switch from the boot loader mode.</p>	<p>boot manual</p> <p>Enables manually booting the switch during the next boot cycle and changes the setting of the MANUAL_BOOT environment variable.</p> <p>The next time you reboot the system, the switch is in boot loader mode. To boot up the system, use the boot flash:filesystem :/file-url boot loader command, and specify the name of the bootable image.</p>

Variable	Boot Loader Command	Cisco IOS Global Configuration Command
CONFIG_FILE	set CONFIG_FILE flash:/ file-url Changes the filename that Cisco IOS uses to read and write a nonvolatile copy of the system configuration.	boot config-file flash:/ file-url Specifies the filename that Cisco IOS uses to read and write a nonvolatile copy of the system configuration. This command changes the CONFIG_FILE environment variable.
BAUD	set BAUD baud-rate	line console 0 speed speed-value Configures the baud rate.
ENABLE_BREAK	set ENABLE_BREAK yes/no	boot enable-break switch yes/no Enables a break to the auto-boot cycle. You have 5 seconds to enter the break command.

Environment Variables for TFTP

When the switch is connected to a PC through the Ethernet management port, you can download or upload a configuration file to the boot loader by using TFTP. Make sure the environment variables in this table are configured.

Table 3: Environment Variables for TFTP

Variable	Description
MAC_ADDR	Specifies the MAC address of the switch. Note We recommend that you do not modify this variable. However, if you modify this variable after the boot loader is up or the value is different from the saved value, enter this command before using TFTP. A reset is required for the new value to take effect.
IP_ADDRESS	Specifies the IP address and the subnet mask for the associated IP subnet of the switch.
DEFAULT_GATEWAY	Specifies the IP address and subnet mask of the default gateway.

How to Perform Device Setup Configuration

Using DHCP to download a new image and a new configuration to a device requires that you configure at least two devices. One device acts as a DHCP and TFTP server and the second device (client) is configured to download either a new configuration file or a new configuration file and a new image file.

Configuring DHCP Autoconfiguration (Only Configuration File)

This task describes how to configure DHCP autoconfiguration of the TFTP and DHCP settings on an existing device in the network so that it can support the autoconfiguration of a new device.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 2	ip dhcp pool <i>poolname</i> Example: Device(config)# <code>ip dhcp pool pool</code>	Creates a name for the DHCP server address pool, and enters DHCP pool configuration mode.
Step 3	boot <i>filename</i> Example: Device(dhcp-config)# <code>boot config-boot.text</code>	Specifies the name of the configuration file that is used as a boot image.
Step 4	network <i>network-number mask prefix-length</i> Example: Device(dhcp-config)# <code>network 10.10.10.0 255.255.255.0</code>	Specifies the subnet network number and mask of the DHCP address pool. Note The prefix length specifies the number of bits that comprise the address prefix. The prefix is an alternative way of specifying the network mask of the client. The prefix length must be preceded by a forward slash (/).
Step 5	default-router <i>address</i> Example: Device(dhcp-config)# <code>default-router 10.10.10.1</code>	Specifies the IP address of the default router for a DHCP client.
Step 6	option 150 <i>address</i> Example: Device(dhcp-config)# <code>option 150 10.10.10.1</code>	Specifies the IP address of the TFTP server.
Step 7	exit Example:	Returns to global configuration mode.

	Command or Action	Purpose
	Device(dhcp-config)# exit	
Step 8	tftp-server flash: <i>filename.text</i> Example: Device(config)# tftp-server flash:config-boot.text	Specifies the configuration file on the TFTP server.
Step 9	interface <i>interface-id</i> Example:	Specifies the address of the client that will receive the configuration file.
Step 10	no switchport Example: Device(config-if)# no switchport	Puts the interface into Layer 3 mode.
Step 11	ip address <i>address mask</i> Example: Device(config-if)# ip address 10.10.10.1 255.255.255.0	Specifies the IP address and mask for the interface.
Step 12	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Configuring DHCP Auto-Image Update (Configuration File and Image)

This task describes DHCP autoconfiguration to configure TFTP and DHCP settings on an existing device to support the installation of a new switch.

Before you begin

You must first create a text file (for example, `autoinstall_dhcp`) that will be uploaded to the device. In the text file, put the name of the image that you want to download (for example, `ess9300_iosxe.17.xx.xx.SPA.bin`).

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example:	Enters global configuration mode.

	Command or Action	Purpose
	Device# <code>configure terminal</code>	
Step 2	ip dhcp pool <i>poolname</i> Example: Device(config)# <code>ip dhcp pool pool1</code>	Creates a name for the DHCP server address pool and enter DHCP pool configuration mode.
Step 3	boot <i>filename</i> Example: Device(dhcp-config)# <code>boot config-boot.text</code>	Specifies the name of the file that is used as a boot image.
Step 4	network <i>network-number mask prefix-length</i> Example: Device(dhcp-config)# <code>network 10.10.10.0 255.255.255.0</code>	Specifies the subnet network number and mask of the DHCP address pool. Note The prefix length specifies the number of bits that comprise the address prefix. The prefix is an alternative way of specifying the network mask of the client. The prefix length must be preceded by a forward slash (/).
Step 5	default-router <i>address</i> Example: Device(dhcp-config)# <code>default-router 10.10.10.1</code>	Specifies the IP address of the default router for a DHCP client.
Step 6	option 150 <i>address</i> Example: Device(dhcp-config)# <code>option 150 10.10.10.1</code>	Specifies the IP address of the TFTP server.
Step 7	option 125 <i>hex</i> Example: Device(dhcp-config)# <code>option 125 hex 0000.0009.0a05.08661.7574.6f69.6e73.7461.6c6c.5f64.686370</code>	Specifies the path to the text file that describes the path to the image file.
Step 8	copy tftp flash <i>filename.txt</i> Example: Device(config)# <code>copy tftp flash image.bin</code>	Uploads the text file to the device.

	Command or Action	Purpose
Step 9	copy tftp flash <i>imagename.bin</i> Example: Device(config)# copy tftp flash image.bin	Uploads the tar file for the new image to the device.
Step 10	exit Example: Device(dhcp-config)# exit	Returns to global configuration mode.
Step 11	tftp-server flash: <i>config.text</i> Example: Device(config)# tftp-server flash:config-boot.text	Specifies the Cisco IOS configuration file on the TFTP server.
Step 12	tftp-server flash: <i>imagename.bin</i> Example: Device(config)# tftp-server flash:image.bin	Specifies the image name on the TFTP server.
Step 13	tftp-server flash: <i>filename.txt</i> Example: Device(config)# tftp-server flash:boot-config.text	Specifies the text file that contains the name of the image file to download
Step 14	interface <i>interface-id</i> Example: Device(config)# interface gigabitEthernet1/0/4	Specifies the address of the client that will receive the configuration file.
Step 15	no switchport Example: Device(config-if)# no switchport	Puts the interface into Layer 3 mode.
Step 16	ip address <i>address mask</i> Example: Device(config-if)# ip address 10.10.10.1	Specifies the IP address and mask for the interface.

	Command or Action	Purpose
	255.255.255.0	
Step 17	end Example: Device(config-if) # end	Returns to privileged EXEC mode.
Step 18	copy running-config startup-config Example: Device(config-if) # end	(Optional) Saves your entries in the configuration file.

Configuring the Client to Download Files from DHCP Server



Note You should only configure and enable the Layer 3 interface. Do not assign an IP address or DHCP-based autoconfiguration with a saved configuration.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	boot host dhcp Example: Device(conf) # boot host dhcp	Enables autoconfiguration with a saved configuration.
Step 3	boot host retry timeout <i>timeout-value</i> Example: Device(conf) # boot host retry timeout 300	(Optional) Sets the amount of time the system tries to download a configuration file. Note If you do not set a timeout, the system will try indefinitely to obtain an IP address from the DHCP server.
Step 4	banner config-save ^C <i>warning-message</i> ^C Example:	(Optional) Creates warning messages to be displayed when you try to save the configuration file to NVRAM.

	Command or Action	Purpose
	Device(conf)# banner config-save ^C Caution - Saving Configuration File to NVRAM May Cause You to No longer Automatically Download Configuration Files at Reboot^C	
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	show boot Example: Device# show boot	Verifies the configuration.

Manually Assigning IP Information to Multiple SVIs

This task describes how to manually assign IP information to multiple switched virtual interfaces (SVIs):

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface vlan <i>vlan-id</i> Example: Device(config)# interface vlan 99	Enters interface configuration mode, and enters the VLAN to which the IP information is assigned. The range is 1 to 4094.
Step 4	ip address <i>ip-address subnet-mask</i> Example: Device(config-vlan)# ip address 10.10.10.2	Enters the IP address and subnet mask.

	Command or Action	Purpose
	255.255.255.0	
Step 5	exit Example: Device(config-vlan)# exit	Returns to global configuration mode.
Step 6	ip default-gateway ip-address Example: Device(config)# ip default-gateway 10.10.10.1	Enters the IP address of the next-hop router interface that is directly connected to the device where a default gateway is being configured. The default gateway receives IP packets with unresolved destination IP addresses from the device. Once the default gateway is configured, the device has connectivity to the remote networks with which a host needs to communicate. Note When your device is configured to route with IP, it does not need to have a default gateway set. Note The device capwap relays on default-gateway configuration to support routed access point join the device.
Step 7	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 8	show interfaces vlan vlan-id Example: Device# show interfaces vlan 99	Displays the interfaces status for the specified VLAN.
Step 9	show ip redirects Example: Device# show ip redirects	Displays the Internet Control Message Protocol (ICMP) redirect messages.

Modifying Device Startup Configuration

The following sections provide information on how to modify the startup configuration of a device.

Specifying a Filename to Read and Write a System Configuration

By default, the Cisco IOS software uses the `config.text` file to read and write a nonvolatile copy of the system configuration. However, you can specify a different filename, which will be loaded during the next boot cycle.

Before you begin

Use a standalone device for this task.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	boot flash:/file-url Example: <pre>Device(config)# boot flash:config.text</pre>	Specifies the configuration file to load during the next boot cycle. <ul style="list-style-type: none"> • <i>file-url</i>: The path (directory) and the configuration filename. • Filenames and directory names are case-sensitive.
Step 4	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 5	show boot Example: <pre>Device# show boot</pre>	Lists the contents of the BOOT environment variable (if set), the name of the configuration file pointed to by the CONFIG_FILE environment variable, and the contents of the BOOTLDR environment variable. <ul style="list-style-type: none"> • The boot global configuration command changes the setting of the CONFIG_FILE environment variable.
Step 6	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring a Scheduled Software Image Reload

This task describes how to configure your device to reload the software image at a later time.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	copy running-config startup-config Example: Device# copy running-config startup-config	Saves your device configuration information to the startup configuration before you use the reload command.
Step 4	reload in [hh:]mm [text] Example: Device# reload in 12 System configuration has been modified. Save? [yes/no]: y	Schedules a reload of the software to take affect in the specified minutes or hours and minutes. The reload must take place within approximately 24 days. You can specify the reason for the reload in a string up to 255 characters in length.
Step 5	reload at hh: mm [month day day month] [text] Example: Device(config)# reload at 14:00	Specifies the time in hours and minutes for the reload to occur. Note Use the at keyword only if the device system clock has been set (through Network Time Protocol (NTP), the hardware calendar, or manually). The time is relative to the configured time zone on the device. To schedule reloads across several devices to occur simultaneously, the time on each device must be synchronized with NTP.
Step 6	reload cancel Example: Device(config)# reload cancel	Cancels a previously scheduled reload.
Step 7	show reload Example:	Displays information about a previously scheduled reload or identifies if a reload has been scheduled on the device.

Command or Action	Purpose
show reload	

Configuration Examples for Device Setup Configuration

The following sections provide configuration examples for device setup.

Examples: Displaying Software Bootup in Install Mode

The following example displays software bootup in install mode:

The following example displays software bootup in bundle mode:

Example: Emergency Installation

The following is a sample output of the **emergency-install** boot command:

```
switch: emergency-install tftp://10.10.0.10/auto/tftpboot/X86/cat9k_iosxe.17.04.01.SPA.bin
WARNING: The system partition (bootflash:) will be erased during the system recovery install
process.
Are you sure you want to proceed? [y] y/n [n]: y
Starting system recovery (tftp://210.10.0.10/auto/tftpboot/X86/cat9k_iosxe.17.04.01.SPA.bin)
...
Attempting to boot from [sda9:cat9k-recovery.SSA.bin]
Located cat9k-recovery.SSA.bin
#####

Warning: ignoring ROMMON var "BOOT_PARAM"

PLATFORM_TYPE 9300 speed 9600

Booting Recovery Image 17.4.1

Initiating Emergency Installation of bundle
tftp://10.10.0.10/auto/tftpboot/X86/cat9k_iosxe.17.04.01.SPA.bin

Downloading bundle tftp://10.10.0.10/auto/tftpboot/X86/cat9k_iosxe.17.04.01.SPA.bin...
curl_vrf=2
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
 100  485M  100  485M    0     0  5143k      0  0:01:36  0:01:36  --:--:-- 5256k
 100  485M  100  485M    0     0  5143k      0  0:01:36  0:01:36  --:--:-- 5143k

Validating bundle tftp://10.10.0.10/auto/tftpboot/X86/cat9k_iosxe.17.04.01.SPA.bin...
Installing bundle tftp://10.10.0.10/auto/tftpboot/X86/cat9k_iosxe.17.04.01.SPA.bin...
Verifying bundle tftp://10.10.0.10/auto/tftpboot/X86/cat9k_iosxe.17.04.01.SPA.bin...
Package cat9k-cc_srdriver.17.04.01.SPA.pkg /temp//stage/cat9k-cc_srdriver.17.04.01.SPA.pkg
is Digitally Signed
Package cat9k-espbase.17.04.01.SPA.pkg /temp//stage/cat9k-espbase.17.04.01.SPA.pkg is
Digitally Signed
Package cat9k-guestshell.17.04.01.SPA.pkg /temp//stage/cat9k-guestshell.17.04.01.SPA.pkg
is Digitally Signed
Package cat9k-rpbase.17.04.01.SPA.pkg /temp//stage/cat9k-rpbase.17.04.01.SPA.pkg is Digitally
Signed
```

```

Package cat9k-sipbase.17.04.01.SPA.pkg /temp//stage/cat9k-sipbase.17.04.01.SPA.pkg is
Digitally Signed
Package cat9k-sipspa.17.04.01.SPA.pkg /temp//stage/cat9k-sipspa.17.04.01.SPA.pkg is Digitally
Signed
Package cat9k-srdriver.17.04.01.SPA.pkg /temp//stage/cat9k-srdriver.17.04.01.SPA.pkg is
Digitally Signed
Package cat9k-webui.17.04.01.SPA.pkg /temp//stage/cat9k-webui.17.04.01.SPA.pkg is Digitally
Signed
Package cat9k-wlc.17.04.01.SPA.pkg /temp//stage/cat9k-wlc.17.04.01.SPA.pkg is Digitally
Signed
Package /cat9k-rpboot.17.04.01.SPA.pkg /temp//rpboot/cat9k-rpboot.17.04.01.SPA.pkg is
Digitally Signed
Preparing flash....
Flash filesystem unmounted successfully /dev/sdb3
Syncing device....
Emergency Install successful... Rebooting
Will reboot now

Initializing Hardware...

System Bootstrap, Version 17.04.01, RELEASE SOFTWARE (P)
Compiled Wed 05/31/2020 15:58:35.22 by rel

Current image running:
Primary Rommon Image

Last reset cause: Software Reload

```

Example: Managing an Update Package

The following example shows how to add a software package file:

```

Device# install add file flash:cat9k_lite_iosxe.17.04.01.SPA.bin activate commit

install_add_activate_commit: START Thu Aug 30 20:25:35 IST 2018

Aug 30 20:25:38.688 IST: %INSTALL-5-INSTALL_START_INFO: Switch 7 R0/0: install_engine:
Started install one-shot flash:cat9k_lite_iosxe.17.04.01.SPA.bininstall_add_activate_commit:
Adding PACKAGE

This operation requires a reload of the system. Do you want to proceed?
Please confirm you have changed boot config to flash:packages.conf [y/n]y

--- Starting initial file syncing ---
[7]: Copying flash:cat9k_lite_iosxe.17.04.01.SPA.bin from switch 7 to switch 4
[4]: Finished copying to switch 4
Info: Finished copying flash:cat9k_lite_iosxe.17.04.01.SPA.bin to the selected switch(es)
Finished initial file syncing

--- Starting Add ---
Performing Add on all members
[4] Add package(s) on switch 4
[4] Finished Add on switch 4
[7] Add package(s) on switch 7
[7] Finished Add on switch 7
Checking status of Add on [4 7]
Add: Passed on [4 7]
Finished Add

install_add_activate_commit: Activating PACKAGE

```

Example: Managing an Update Package

```

gzip: initramfs.cpio.gz: decompression OK, trailing garbage ignored
Following packages shall be activated:
/flash/cat9k_lite-webui.17.04.01.SPA.pkg
/flash/cat9k_lite-srdriver.17.04.01.SPA.pkg
/flash/cat9k_lite-rpboot.17.04.01.SPA.pkg
/flash/cat9k_lite-rpbase.17.04.01.SPA.pkg

This operation requires a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on all members

Aug 30 20:51:16.365 IST: %INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: Switch 7 R0/0:
rollback_timer: Install auto abort timer will expire in 7200 seconds [4] Activate package(s)
on switch 4
[4] Finished Activate on switch 4
[7] Activate package(s) on switch 7

Aug 30 20:51:17.561 IST: %INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: Switch 4 R0/0:
rollback_timer: Install auto abort timer will expire in 7200 seconds [7] Finished Activate
on switch 7
Checking status of Activate on [4 7]
Activate: Passed on [4 7]
Finished Activate

--- Starting Commit ---
Performing Commit on all members
[4] Commit package(s) on switch 4
[4] Finished Commit on switch 4
[7] Commit package(s) on switch 7
[7] Finished Commit on switch 7
Checking status of Commit on [4 7]
Commit: Passed on [4 7]
Finished Commit

Install will reload the system now!
SUCCESS: install_add_activate_commit Thu Aug 30 20:51:55 IST 2018

Y2#
Chassis 7 reloading, reason - Reload command

Aug 30 20:51:56.017 IST: %INSTALL-5-INSTALL_COMPLETED_INFO: Switch 7 R0/0: install_engine:
Completed install one-shot PACKAGE flash:cat9k_lite_iosxe.17.04.01.SPA.binAug 30
20:52:03.517: %PMAN-5-EXITACTION: F0/0: pvp: Process manager is exiting: reload fp action
requested
Aug 30 20:52:07.543: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting: rp processes
exit with reload switch code

Aug 30 20:52:11.104: %PMAN-5-EXITACTION: C0/0: pvp: Process manager is exiting: reload cc
action requested
reboot: Restarting system

```

The following is a sample output of the **show install summary** command after adding a software package file to a device:

```

Device# show install summary
[ Switch 4 7 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
           C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St   Filename/Version
-----
IMG   C    16.9.1.0.70
-----

```

```
Auto abort timer: inactive
-----
```

The following example shows how to activate an added software package file:

```
Device# install activate
```

```
install_activate: START Mon Oct 30 20:14:20 UTC 2017
install_activate: Activating PACKAGE
```

```
*Oct 30 20:14:21.379: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:14:21 install_engine.sh:
```

```
%INSTALL-5-INSTALL_START_INFO: Started install activateFollowing packages shall be activated:
/flash/cat9k-wlc.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-webui.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-sipspace.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-rpboot.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-guestshell.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-espspace.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
```

```
This operation requires a reload of the system. Do you want to proceed? [y/n]
```

```
--- Starting Activate ---
```

```
Performing Activate on all members
```

```
[1] Activate package(s) on switch 1
```

```
--- Starting list of software package changes ---
```

```
Old files list:
```

```
Removed cat9k-cc_srdriver.16.06.02.SPA.pkg
Removed cat9k-espspace.16.06.02.SPA.pkg
Removed cat9k-guestshell.16.06.02.SPA.pkg
Removed cat9k-rpbase.16.06.02.SPA.pkg
Removed cat9k-rpboot.16.06.02.SPA.pkg
Removed cat9k-sipbase.16.06.02.SPA.pkg
Removed cat9k-sipspace.16.06.02.SPA.pkg
Removed cat9k-srdriver.16.06.02.SPA.pkg
Removed cat9k-webui.16.06.02.SPA.pkg
Removed cat9k-wlc.16.06.02.SPA.pkg
```

```
New files list:
```

```
Added cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Added cat9k-espspace.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Added cat9k-guestshell.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Added cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Added cat9k-rpboot.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Added cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Added cat9k-sipspace.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Added cat9k-srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Added cat9k-webui.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Added cat9k-wlc.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
```

```
Finished list of software package changes
```

```
[1] Finished Activate on switch 1
```

```
Checking status of Activate on [1]
```

```
Activate: Passed on [1]
```

```
Finished Activate
```

```
*Oct 30 20:15:56.572: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:15:56 rollback_timer.sh:
```

```
%INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: Install auto abort timer will expire in 7200
seconds
```

```
Install will reload the system now!
```

```
SUCCESS: install_activate Mon Oct 30 20:16:01 UTC 2017
```

```
Device#
*Oct 30 20:16:01.935: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:16:01
install_engine.sh: %INSTALL-5-INSTALL_COMPLETED_INFO: Completed install activate PACKAGE
Chassis 1 reloading, reason - Reload command
```

The following sample output from the **show install summary** command displays the status of the software package as active and uncommitted:

```
Device# show install summary
```

```
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
           C - Activated & Committed, D - Deactivated & Uncommitted
```

```
-----
Type  St  Filename/Version
-----
```

```
IMG   I   16.6.2.0
IMG   U   16.6.1.0
```

```
Device#
```

The following example shows how to execute the **install commit** command:

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

The following is a sample output from the **install remove inactive** command:

The following is sample output from the **install abort** command:

The following is a sample output from the **install activate auto-abort-timer** command:

Verifying Software Install

Procedure

Step 1 enable

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 show install log

Example:

```
Device# show install log
```

Displays information about all the software install operations that was performed since boot-up of the device.

Step 3 show install summary

Example:

```
Device# show install summary
```

Displays information about the image versions and their corresponding install state for all members/field-replaceable unit (FRU).

- The output of this command differs based on the **install** command that is executed.

Step 4 **show install package** *filesystem: filename*

Example:

```
Device# show install package flash:cat9k_lite-rpboot.17.04.01.SPA.pkg
```

Displays information about the specified software install package file.

```
Device# show install package flash:cat9k_lite-rpboot.17.04.01.SPA.pkg
Package: cat9k_lite-rpboot.17.04.01.SPA.pkg
Size: 34616705
Timestamp: Thu Aug 30 20:28:25 2018 UTC
Canonical path: /flash/cat9k_lite-rpboot.17.04.01.SPA.pkg
```

```
Raw disk-file SHA1sum:
 5e816f97bcae3e30eb8bc2f0ec8f64402ceal638
Header size:      980 bytes
Package type:    30001
Package flags:   0
Header version:  3
```

Package is bootable on RP when specified by packages provisioning file.

Step 5 **show install active**

Example:

```
Device# show install active
```

Step 6 **show install inactive**

Example:

```
Device# show install inactive
```

Displays information about the inactive packages.

```
Device# show install inactive
```

```
[ Switch 1 2 ] Inactive Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
             C - Activated & Committed, D - Deactivated & Uncommitted
```

```
-----
Type  St  Filename/Version
-----
```

```
IMG  I   16.9.2.0
```

```
Device#
```

Step 7 **show install committed**

Example:

```
Device# show install committed
```

Displays information about committed packages.

Example: Configuring a Device to Download Configurations from a DHCP Server

```
Device# show install committed

[ Switch 1 2 ] Committed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
           C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St  Filename/Version
-----
IMG   C   16.9.1.0
Device#
```

Step 8 show install uncommitted

Example:

```
Device# show install uncommitted
```

Displays information about uncommitted packages.

```
Device# show install uncommitted

[ Switch 1 2 ] Uncommitted Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
           C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St  Filename/Version
-----
IMG   U   16.9.2.0
Device#
```

Example: Configuring a Device to Download Configurations from a DHCP Server

The following example shows how to use a Layer 3 SVI interface on VLAN 99 to enable DHCP-based autoconfiguration with a saved configuration:

```
Device# configure terminal
Device(config)# boot host dhcp
Device(config)# boot host retry timeout 300
Device(config)# banner config-save ^C Caution - Saving Configuration File to NVRAM May Cause
  You to No longer Automatically Download Configuration Files at Reboot^C
Device(config)# vlan 99
Device(config-vlan)# interface vlan 99
Device(config-if)# no shutdown
Device(config-if)# end
Device# show boot

BOOT path-list:
Config file:          flash:/config.text
Private Config file:  flash:/private-config.text
Enable Break:         no
Manual Boot:          no
HELPER path-list:
NVRAM/Config file
  buffer size:        32768
Timeout for Config
  Download:           300 seconds
```

```
Config Download
  via DHCP:      enabled (next boot: enabled)
Device#
```

Example: Scheduling Software Image Reload

This example shows how to reload the software on a device on the current day at 7:30 p.m:

```
Device# reload at 19:30

Reload scheduled for 19:30:00 UTC Wed Jun 5 2013 (in 2 hours and 25 minutes)
Proceed with reload? [confirm]
```

This example shows how to reload the software on a device at a future date and time:

```
Device# reload at 02:00 jun 20

Reload scheduled for 02:00:00 UTC Thu Jun 20 2013 (in 344 hours and 53 minutes)
Proceed with reload? [confirm]
```

