



Performing Controller Setup Configuration

- [Information About Performing Controller Setup Configuration, page 1](#)
- [How to Perform Controller Setup Configuration, page 9](#)
- [Monitoring Controller Setup Configuration, page 17](#)
- [Configuration Examples for Performing Controller Setup, page 19](#)
- [Feature History and Information For Performing Controller Setup Configuration, page 20](#)

Information About Performing Controller Setup Configuration

Review the sections in this module before performing your initial controller configuration tasks that include IP address assignments and DHCP autoconfiguration.

Controller Boot Process

To start your controller, you need to follow the procedures in the hardware installation guide for installing and powering on the controller and setting up the initial controller configuration (IP address, subnet mask, default gateway, secret and Telnet passwords, and so forth).

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

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

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.

The boot loader also provides trap-door access into the system if the operating system has problems serious enough that it cannot be used. The trap-door mechanism provides enough access to the system so that if it is

necessary, you can reinstall the operating system software image by using the Xmodem Protocol, recover from a lost or forgotten password, and finally restart the operating system.

Before you can assign controller 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 controller 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.

Software Installer Features

The following software installer features are supported on your switch:

- Software bundle installation on a standalone switch.
- In a stack of switches, Cisco recommends all switches in install mode.
- Software rollback to a previously installed package set.
- Emergency installation in the event that no valid installed packages reside on the boot flash.



Note

Software installation and rollback must be performed while running only in installed mode. You can use the **software expand EXEC** command to convert bundle boot mode to install mode.

Software Boot Modes

Your controller supports two modes to boot the software packages:

- Installed mode
- Bundle mode

Related Topics

[Examples: Displaying Software Bootup in Install Mode, on page 18](#)

[Example: Emergency Installation, on page 19](#)

Installed Boot Mode

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

```
controller: boot flash:packages.conf
```

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.

Related Topics

[Examples: Displaying Software Bootup in Install Mode, on page 18](#)

[Example: Emergency Installation, on page 19](#)

Bundle Boot Mode

You can boot your controller 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.

**Note**

Auto install and smart install functionality is not supported in bundle boot mode.

Related Topics

[Examples: Displaying Software Bootup in Install Mode, on page 18](#)

[Example: Emergency Installation, on page 19](#)

Controllers Information Assignment

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

Use the controller 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.

The switch stack is managed through a single IP address. The IP address is a system-level setting and is not specific to the stack master or to any other stack member. You can still manage the stack through the same

IP address even if you remove the stack master or any other stack member from the stack, provided there is IP connectivity.

**Note**

Stack members retain their IP address when you remove them from a switch stack. To avoid a conflict by having two devices with the same IP address in your network, change the IP address of the switch that you removed from the switch stack.

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 controller receives the dynamically assigned IP address and reads the configuration file.

If you are an experienced user familiar with the controller configuration steps, manually configure the controller. Otherwise, use the setup program described in the *Boot Process* section.

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 controller can act as both a DHCP client and a DHCP server.

During DHCP-based autoconfiguration, your controller (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 controller. 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 controller can be on the same LAN or on a different LAN than the controller. If the DHCP server is running on a different LAN, you should configure a DHCP relay device between your controller 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 controller.

DHCP Client Request Process

When you boot up your controller, the DHCP client is invoked and requests configuration information from a DHCP server when the configuration file is not present on the controller. 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 1: DHCP Client and Server Message Exchange

The client, Controller 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 controller 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 controller accepts replies from a BOOTP server and configures itself, the controller broadcasts, instead of unicasts, TFTP requests to obtain the controller configuration file.

The DHCP hostname option allows a group of controllers to obtain hostnames and a standard configuration from the central management DHCP server. A client (controller) 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.

If a client has a default hostname (the **hostname name** global configuration command is not configured or the **no hostname** global configuration command is entered to remove the hostname), the DHCP hostname option is not included in the packet when you enter the **ip address dhcp** interface configuration command. In this case, if the client receives the DHCP hostname option from the DHCP interaction while acquiring an IP address for an interface, the client accepts the DHCP hostname option and sets the flag to show that the system now has a hostname configured.

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 controller by the controller hardware address.
- If you want the controller 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 controller) (required)
- If you want the controller 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 controller 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 controller is not configured. If the router IP address or the TFTP server name are not found, the controller might send broadcast, instead of unicast, TFTP requests. Unavailability of other lease options does not affect autoconfiguration.
- The controller can act as a DHCP server. By default, the Cisco IOS DHCP server and relay agent features are enabled on your controller but are not configured. (These features are not operational.)

Purpose of the TFTP Server

Based on the DHCP server configuration, the controller attempts to download one or more configuration files from the TFTP server. If you configured the DHCP server to respond to the controller 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 controller 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 controller 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 controller'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 controller 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 controller 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 controllers. 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 controller, or if it is to be accessed by the controller 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 controller.

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 controller. If it is on a different LAN, the controller 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 controller obtains its configuration information in these ways:

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

The controller receives its IP address, subnet mask, TFTP server address, and the configuration filename from the DHCP server. The controller 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 controller, but the TFTP server address is not provided in the DHCP reply (one-file read method).

The controller receives its IP address, subnet mask, and the configuration filename from the DHCP server. The controller 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 controller and provided in the DHCP reply. The configuration filename is not provided (two-file read method).

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

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

After obtaining its hostname from the default configuration file or the DHCP reply, the controller reads the configuration file that has the same name as its hostname (*hostname-conf* or *hostname.cfg*, depending on whether network-conf 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 controller cannot read the network-conf, cisco.net.cfg, or the hostname file, it reads the router-conf file. If the controller cannot read the router-conf file, it reads the ciscotr.cfg file.

**Note**

The controller 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.

How to Control Environment Variables

With a normally operating controller, you enter the boot loader mode only through the console connection configured for 9600 bps. Unplug the controller power cord, and press the **Mode** button while reconnecting the power cord. You can release the **Mode** button after all the amber system LEDs turn on and remain solid. The boot loader controller prompt then appears.

The controller 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.

Scheduled Reload of the Software Image

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

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

If you modify your configuration file, the controller 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 Perform Controller Setup Configuration

Using DHCP to download a new image and a new configuration to a controller requires that you configure at least two controllers. One controller acts as a DHCP and TFTP server and the second controller (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 controller in the network so that it can support the autoconfiguration of a new controller.

SUMMARY STEPS

1. **configure terminal**
2. **ip dhcp pool** *poolname*
3. **boot** *filename*
4. **network** *network-number mask prefix-length*
5. **default-router** *address*
6. **option 150** *address*
7. **exit**
8. **tftp-server flash:***filename.text*
9. **interface** *interface-id*
10. **no switchport**
11. **ip address** *address mask*
12. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	ip dhcp pool <i>poolname</i> Example: Controller(config)# ip dhcp pool pool	Creates a name for the DHCP server address pool, and enters DHCP pool configuration mode.
Step 3	boot <i>filename</i> Example: Controller(dhcp-config)# boot config-boot.text	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: Controller(dhcp-config)# network 10.10.10.0 255.255.255.0	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: Controller(dhcp-config)# default-router 10.10.10.1	Specifies the IP address of the default router for a DHCP client.
Step 6	option 150 <i>address</i> Example: Controller(dhcp-config)# option 150 10.10.10.1	Specifies the IP address of the TFTP server.
Step 7	exit Example: Controller(dhcp-config)# exit	Returns to global configuration mode.
Step 8	tftp-server flash: <i>filename.text</i> Example: Controller(config)# tftp-server flash:config-boot.text	Specifies the configuration file on the TFTP server.

	Command or Action	Purpose
Step 9	interface <i>interface-id</i> Example: Controller(config) # interface gigabitethernet1/0/4	Specifies the address of the client that will receive the configuration file.
Step 10	no switchport Example: Controller(config-if) # no switchport	Puts the interface into Layer 3 mode.
Step 11	ip address <i>address mask</i> Example: Controller(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: Controller(config-if) # end	Returns to privileged EXEC mode.

Manually Assigning IP Information to Multiple SVIs

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

SUMMARY STEPS

1. **configure terminal**
2. **interface vlan** *vlan-id*
3. **ip address** *ip-address subnet-mask*
4. **exit**
5. **ip default-gateway** *ip-address*
6. **end**
7. **show interfaces vlan** *vlan-id*
8. **show ip redirects**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Controller# configure terminal	Enters global configuration mode.
Step 2	interface vlan <i>vlan-id</i> Example: Controller(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 3	ip address <i>ip-address subnet-mask</i> Example: Controller(config-vlan)# ip address 10.10.10.2 255.255.255.0	Enters the IP address and subnet mask.
Step 4	exit Example: Controller(config-vlan)# exit	Returns to global configuration mode.
Step 5	ip default-gateway <i>ip-address</i> Example: Controller(config)# ip default-gateway 10.10.10.1	<p>Enters the IP address of the next-hop router interface that is directly connected to the controller where a default gateway is being configured. The default gateway receives IP packets with unresolved destination IP addresses from the controller.</p> <p>Once the default gateway is configured, the controller has connectivity to the remote networks with which a host needs to communicate.</p> <p>Note When your controller is configured to route with IP, it does not need to have a default gateway set.</p> <p>Note The controller capwap relays on default-gateway configuration to support routed access point join the controller.</p>
Step 6	end Example: Controller(config)# end	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 7	show interfaces vlan <i>vlan-id</i> Example: Controller# show interfaces vlan 99	Verifies the configured IP address.
Step 8	show ip redirects Example: Controller# show ip redirects	Verifies the configured default gateway.

Modifying the Controller Startup Configuration

Specifying the Filename to Read and Write the 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 controller for this task.

SUMMARY STEPS

1. **configure terminal**
2. **boot flash:/file-url**
3. **end**
4. **show boot**
5. **copy running-config startup-config**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	boot flash: <i>file-url</i> Example: Controller(config)# boot flash:config.text	Specifies the configuration file to load during the next boot cycle. <i>file-url</i> —The path (directory) and the configuration filename. Filenames and directory names are case-sensitive.
Step 3	end Example: Controller(config)# end	Returns to privileged EXEC mode.
Step 4	show boot Example: Controller# show boot	Verifies your entries. The boot global configuration command changes the setting of the CONFIG_FILE environment variable.
Step 5	copy running-config startup-config Example: Controller# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Booting the Controller in Installed Mode

SUMMARY STEPS

1. **cp** *source_file_path destination_file_path*
2. **software expand file** *source_file_path*
3. **reload**
4. **boot flash:packages.conf**
5. **show version**

DETAILED STEPS

	Command or Action	Purpose
Step 1	cp <i>source_file_path destination_file_path</i> Example: Controller#	(Optional) Copies the bin file (image.bin) from the FTP or TFTP server to flash or USB flash.

	Command or Action	Purpose
Step 2	software expand file <i>source_file_path</i> Example: Controller: software expand file <i>source_file_path</i>	Expands the bin file stored in flash, FTP, TFTP, HTTP, or HTTPS server on the booted controller. Note Ensure that the <code>packages.conf</code> file is available in the expanded list.
Step 3	reload Example: Controller: reload	Reloads the controller. Note You can boot the controller manually or automatically using the <code>packages.conf</code> file. If you are booting manually, you can proceed to Step 4. Otherwise, the controller boots up automatically.
Step 4	boot flash:packages.conf Example: switch: boot flash:packages.conf	Boots the controller with the <code>packages.conf</code> file.
Step 5	show version Example:	Verifies that the controller is in the INSTALL mode.

Booting the Controller in Bundle Mode

There are several methods by which you can boot the controller—either by copying the bin file from the TFTP server and then boot the controller, or by booting the controller 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 controller from the TFTP server in the bundle mode.

SUMMARY STEPS

1. **cp** *source_file_path destination_file_path*
- 2.
3. **boot**
4. **show version**

DETAILED STEPS

	Command or Action	Purpose
Step 1	cp <i>source_file_path destination_file_path</i> Example: Controller#	(Optional) Copies the bin file (<code>image.bin</code>) from the FTP or TFTP server to flash or USB flash.

	Command or Action	Purpose
Step 2	Example: Controller:	Sets the boot parameters.
Step 3	boot Example: switch: boot	Boots the controller.
Step 4	show version Example:	Verifies that the controller is in the BUNDLE mode.

Configuring a Scheduled Software Image Reload

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

SUMMARY STEPS

1. **configure terminal**
2. **copy running-config startup-config**
3. **reload in [hh:]mm [text]**
4. **reload at hh: mm [month day | day month] [text]**
5. **reload cancel**
6. **show reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Controller# configure terminal	Enters global configuration mode.
Step 2	copy running-config startup-config Example: copy running-config startup-config	Saves your controller configuration information to the startup configuration before you use the reload command.
Step 3	reload in [hh:]mm [text] Example: Controller(config)# reload in 12	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.

	Command or Action	Purpose
	System configuration has been modified. Save? [yes/no]: y	
Step 4	reload at <i>hh: mm [month day day month]</i> <i>[text]</i> Example: Controller(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 controller 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 controller. To schedule reloads across several controllers to occur simultaneously, the time on each controller must be synchronized with NTP.
Step 5	reload cancel Example: Controller(config)# reload cancel	Cancels a previously scheduled reload.
Step 6	show reload Example: show reload	Displays information about a previously scheduled reload or identifies if a reload has been scheduled on the controller.

Monitoring Controller Setup Configuration

Example: Verifying the Controller Running Configuration

```

Controller# show running-config
Building configuration...

Current configuration: 1363 bytes
!
version 12.4
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Stack1
!
enable secret 5 $1$ej9.$DMUvAUUnZOAmvmgqBEzIxE0
!
.
<output truncated>
.
interface gigabitethernet6/0/2
mvr type source

<output truncated>

...!
interface VLAN1

```

```

ip address 172.20.137.50 255.255.255.0
no ip directed-broadcast
!
ip default-gateway 172.20.137.1 !
!
snmp-server community private RW
snmp-server community public RO
snmp-server community private@es0 RW
snmp-server community public@es0 RO
snmp-server chassis-id 0x12
!
end

```

Examples: Displaying Software Bootup in Install Mode

This example displays software bootup in install mode:

switch: **boot flash:packages.conf**

```

Getting rest of image
Reading full image into memory...done
Reading full base package into memory...: done = 74596432
Nova Bundle Image
-----
Kernel Address : 0x6042f354
Kernel Size : 0x318412/3245074
Initramfs Address : 0x60747768
Initramfs Size : 0xdc08e8/14420200
Compression Format: .mzip

Bootable image at @ ram:0x6042f354
Bootable image segment 0 address range [0x81100000, 0x81b80000] is in range [0x80180000,
0x900000000].
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@boot_system:
377
Loading Linux kernel with entry point 0x811060f0 ...
Bootloader: Done loading app on core_mask: 0xf

### Launching Linux Kernel (flags = 0x5)

All packages are Digitally Signed
Starting System Services
Nov 7 09:57:05 %IOSXE-1-PLATFORM: process stack-mgr: %STACKMGR-1-DISC_START: Switch 2 is
starting stack discovery
#####
Nov 7 09:59:07 %IOSXE-1-PLATFORM: process stack-mgr: %STACKMGR-1-DISC_DONE: Switch 2 has
finished stack discovery
Nov 7 09:59:07 %IOSXE-1-PLATFORM: process stack-mgr: %STACKMGR-1-SWITCH_ADDED: Switch 2 has
been added to the stack
Nov 7 09:59:14 %IOSXE-1-PLATFORM: process stack-mgr: %STACKMGR-1-ACTIVE_ELECTED: Switch 2
has been elected 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 L3 Switch Software (CAT3K_CAA-UNIVERSALK9-M),
Version 03.09.12.EMD EARLY DEPLOYMENT ENGINEERING NOVA_WEEKLY BUILD, synced to

```

```

DSGS_PI2_POSTPC_FLO_DSBUE7_NG3K_1105
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Sun 04-Nov-12 22:53 by gereddy
License level to iosd is ipservices

```

Related Topics

[Software Boot Modes, on page 2](#)

[Installed Boot Mode, on page 3](#)

[Bundle Boot Mode, on page 3](#)

Example: Emergency Installation

This sample output is an example when the **emergency-install** boot command is initiated:

Related Topics

[Software Boot Modes, on page 2](#)

[Installed Boot Mode, on page 3](#)

[Bundle Boot Mode, on page 3](#)

Configuration Examples for Performing Controller Setup

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

This example uses a Layer 3 SVI interface on VLAN 99 to enable DHCP-based autoconfiguration with a saved configuration:

```

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

```

Controller#

Examples: Scheduling Software Image Reload

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

```
Controller# 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 the controller at a future time:

```
Controller# 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]
```

Feature History and Information For Performing Controller Setup Configuration

Command History

Release	Modification
Cisco IOS XE 3.2SE	This feature was introduced.