Cisco IOS Configuration Fundamentals Configuration Guide

Release 12.4

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

About Cisco IOS and Cisco IOS XE Software Documentation

Last updated: August 6, 2008

This document describes the objectives, audience, conventions, and organization used in Cisco IOS and Cisco IOS XE software documentation, collectively referred to in this document as Cisco IOS documentation. Also included are resources for obtaining technical assistance, additional documentation, and other information from Cisco. This document is organized into the following sections:

Documentation Objectives

Cisco IOS documentation describes the tasks and commands available to configure and maintain Cisco networking devices.

Audience

The Cisco IOS documentation set is i ntended for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the configuration and maintenance tasks, the relationship among tasks, or the Cisco IOS commands necessary to perform particular tasks. The Cisco IOS documentation set is also intended for those users experienced with Cisco IOS who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS release.

Documentation Conventions

In Cisco IOS documentation, the term *router* may be used to refer to various Cisco products; for example, routers, access servers, and switches. These and other networking devices that support Cisco IOS software are shown interchangeably in examples and are used only for illustrative purposes. An example that shows one product does not necessarily mean that other products are not supported.

This section includes the following topics:

- Typographic Conventions, page ii
- Command Syntax Conventions, page ii
- Software Conventions, page iii
- Reader Alert Conventions, page iii

Typographic Conventions

Cisco IOS documentation uses the following typographic conventions:

Convention	Description
^ or Ctrl	Both the ^ symbol and Ctrl represent the Control (Ctrl) key on a keyboard. For example, the key combination **^D** or **Ctrl-D** means that you hold down the Control key while you press the D key. (Keys are indicated in capital letters but are not case sensitive.)
string	A string is a nonquoted set of characters shown in italics. For example, when setting a Simple Network Management Protocol (SNMP) community string to *public*, do not use quotation marks around the string; otherwise, the string will include the quotation marks.

Command Syntax Conventions

Cisco IOS documentation uses the following command syntax conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter as shown.
italic	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional keyword or argument.
|	A vertical line, called a pipe, indicates a choice within a set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a pipe indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a pipe indicate a required choice.
[x {y | z}]	Braces and a pipe within square brackets indicate a required choice within an optional element.

Software Conventions

Cisco IOS uses the following program code conventions:

Convention	Description
`Courier font`	Courier font is used for information that is displayed on a PC or terminal screen.
`Bold Courier font`	Bold Courier font indicates text that the user must enter.
< >	Angle brackets enclose text that is not displayed, such as a password. Angle brackets also are used in contexts in which the italic font style is not supported; for example, ASCII text.
!	An exclamation point at the beginning of a line indicates that the text that follows is a comment, not a line of code. An exclamation point is also displayed by Cisco IOS software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Reader Alert Conventions

The Cisco IOS documentation set uses the following conventions for reader alerts:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Documentation Organization

This section describes the Cisco IOS documentation set, how it is organized, and how to access it on Cisco.com. Included are lists of configuration guides, command references, and supplementary references and resources that make up the documentation set. The following topics are included:

- Cisco IOS Documentation Set, page iv
- Cisco IOS Documentation on Cisco.com, page iv
- Configuration Guides, Command References, and Supplementary Resources, page v

Cisco IOS Documentation Set

Cisco IOS documentation consists of the following:

- Release notes and caveats provide information about platform, technology, and feature support for a release and describe severity 1 (catastrophic), severity 2 (severe), and severity 3 (moderate) defects in released Cisco IOS code. Review release notes before other documents to learn whether or not updates have been made to a feature.
- Sets of configuration guides and command references organized by technology and published for each standard Cisco IOS release.
  - Configuration guides—Compilations of documents that provide informational and task-oriented descriptions of Cisco IOS features.
  - Command references—Compilations of command pages that provide detailed information about the commands used in the Cisco IOS features and processes that make up the related configuration guides. For each technology, there is a single command reference that covers all Cisco IOS releases and that is updated at each standard release.
- Lists of all the commands in a specific release and all commands that are new, modified, removed, or replaced in the release.
- Command reference book for **debug** commands. Command pages are listed in alphabetical order.
- Reference book for system messages for all Cisco IOS releases.

Cisco IOS Documentation on Cisco.com

The following sections describe the documentation organization and how to access various document types.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

New Features List

The New Features List for each release provides a list of all features in the release with hyperlinks to the feature guides in which they are documented.

Feature Guides

Cisco IOS features are documented in feature guides. Feature guides describe one feature or a group of related features that are supported on many different software releases and platforms. Your Cisco IOS software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release.

Configuration Guides

Configuration guides are provided by technology and release and comprise a set of individual feature guides relevant to the release and technology.

Command References

Command reference books describe Cisco IOS commands that are supported in many different software releases and on many different platforms. The books are provided by technology. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

Cisco IOS Supplementary Documents and Resources

Supplementary documents and resources are listed in Table 2 on page xi.

Configuration Guides, Command References, and Supplementary Resources

Table 1 lists, in alphabetical order, Cisco IOS and Cisco IOS XE software configuration guides and command references, including brief descriptions of the contents of the documents. The Cisco IOS command references are comprehensive, meaning that they include commands for both Cisco IOS software and Cisco IOS XE software, for all releases. The configuration guides and command references support many different software releases and platforms. Your Cisco IOS software release or platform may not support all these technologies.

For additional information about configuring and operating specific networking devices, go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

Table 2 lists documents and resources that supplement the Cisco IOS software configuration guides and command references. These supplementary resources include release notes and caveats; master command lists; new, modified, removed, and replaced command lists; system messages; and the debug command reference.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS AppleTalk Configuration Guide *Cisco IOS XE AppleTalk Configuration Guide* *Cisco IOS AppleTalk Command Reference*	AppleTalk protocol.
Cisco IOS Asynchronous Transfer Mode Configuration Guide *Cisco IOS Asynchronous Transfer Mode Command Reference*	LAN ATM, multiprotocol over ATM (MPoA), and WAN ATM.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Bridging and IBM Networking Configuration Guide *Cisco IOS Bridging Command Reference* *Cisco IOS IBM Networking Command Reference*	• Transparent and source-route transparent (SRT) bridging, source-route bridging (SRB), Token Ring Inter-Switch Link (TRISL), and token ring route switch module (TRRSM). • Data-link switching plus (DLSw+), serial tunnel (STUN), block serial tunnel (BSTUN); logical link control, type 2 (LLC2), synchronous data link control (SDLC); IBM Network Media Translation, including Synchronous Data Logical Link Control (SDLLC) and qualified LLC (QLLC); downstream physical unit (DSPU), Systems Network Architecture (SNA) service point, SNA frame relay access, advanced peer-to-peer networking (APPN), native client interface architecture (NCIA) client/server topologies, and IBM Channel Attach.
Cisco IOS Broadband and DSL Configuration Guide *Cisco IOS XE Broadband and DSL Configuration Guide* *Cisco IOS Broadband and DSL Command Reference*	Point-to-Point Protocol (PPP) over ATM (PPPoA) and PPP over Ethernet (PPPoE).
Cisco IOS Carrier Ethernet Configuration Guide *Cisco IOS Carrier Ethernet Command Reference*	Connectivity fault management (CFM), Ethernet Local Management Interface (ELMI), IEEE 802.3ad link bundling, Link Layer Discovery Protocol (LLDP), media endpoint discovery (MED), and operations, administration, and maintenance (OAM).
Cisco IOS Configuration Fundamentals Configuration Guide *Cisco IOS XE Configuration Fundamentals Configuration Guide* *Cisco IOS Configuration Fundamentals Command Reference*	Autoinstall, Setup, Cisco IOS command-line interface (CLI), Cisco IOS file system (IFS), Cisco IOS web browser user interface (UI), basic file transfer services, and file management.
Cisco IOS DECnet Configuration Guide *Cisco IOS XE DECnet Configuration Guide* *Cisco IOS DECnet Command Reference*	DECnet protocol.
Cisco IOS Dial Technologies Configuration Guide *Cisco IOS XE Dial Technologies Configuration Guide* *Cisco IOS Dial Technologies Command Reference*	Asynchronous communications, dial backup, dialer technology, dial-in terminal services and AppleTalk remote access (ARA), large scale dialout, dial-on-demand routing, dialout, modem and resource pooling, ISDN, multilink PPP (MLP), PPP, virtual private dialup network (VPDN).
Cisco IOS Flexible NetFlow Configuration Guide *Cisco IOS Flexible NetFlow Command Reference*	Flexible NetFlow.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS H.323 Configuration Guide	Gatekeeper enhancements for managed voice services, Gatekeeper Transaction Message Protocol, gateway codec order preservation and shutdown control, H.323 dual tone multifrequency relay, H.323 version 2 enhancements, Network Address Translation (NAT) support of H.323 v2 Registration, Admission, and Status (RAS) protocol, tokenless call authorization, and VoIP gateway trunk and carrier-based routing.
Cisco IOS High Availability Configuration Guide *Cisco IOS XE High Availability Configuration Guide* *Cisco IOS High Availability Command Reference*	A variety of High Availability (HA) features and technologies that are available for different network segments (from enterprise access to service provider core) to facilitate creation of end-to-end highly available networks. Cisco IOS HA features and technologies can be categorized in three key areas: system-level resiliency, network-level resiliency, and embedded management for resiliency.
Cisco IOS Integrated Session Border Controller Command Reference	A VoIP-enabled device that is deployed at the edge of networks. An SBC is a toolkit of functions, such as signaling interworking, network hiding, security, and quality of service (QoS).
Cisco IOS Intelligent Service Gateway Configuration Guide *Cisco IOS Intelligent Service Gateway Command Reference*	Subscriber identification, service and policy determination, session creation, session policy enforcement, session life-cycle management, accounting for access and service usage, session state monitoring.
Cisco IOS Interface and Hardware Component Configuration Guide *Cisco IOS XE Interface and Hardware Component Configuration Guide* *Cisco IOS Interface and Hardware Component Command Reference*	LAN interfaces, logical interfaces, serial interfaces, virtual interfaces, and interface configuration.
Cisco IOS IP Addressing Services Configuration Guide *Cisco IOS XE Addressing Services Configuration Guide* *Cisco IOS IP Addressing Services Command Reference*	Address Resolution Protocol (ARP), Network Address Translation (NAT), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Next Hop Address Resolution Protocol (NHRP).
Cisco IOS IP Application Services Configuration Guide *Cisco IOS XE IP Application Services Configuration Guide* *Cisco IOS IP Application Services Command Reference*	Enhanced Object Tracking (EOT), Gateway Load Balancing Protocol (GLBP), Hot Standby Router Protocol (HSRP), IP Services, Server Load Balancing (SLB), Stream Control Transmission Protocol (SCTP), TCP, Web Cache Communication Protocol (WCCP), User Datagram Protocol (UDP), and Virtual Router Redundancy Protocol (VRRP).
Cisco IOS IP Mobility Configuration Guide *Cisco IOS IP Mobility Command Reference*	Mobile ad hoc networks (MANet) and Cisco mobile networks.
Cisco IOS IP Multicast Configuration Guide *Cisco IOS XE IP Multicast Configuration Guide* *Cisco IOS IP Multicast Command Reference*	Protocol Independent Multicast (PIM) sparse mode (PIM-SM), bidirectional PIM (bidir-PIM), Source Specific Multicast (SSM), Multicast Source Discovery Protocol (MSDP), Internet Group Management Protocol (IGMP), and Multicast VPN (MVPN).

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS IP Routing Protocols Configuration Guide *Cisco IOS XE IP Routing Protocols Configuration Guide* *Cisco IOS IP Routing Protocols Command Reference*	Border Gateway Protocol (BGP), multiprotocol BGP, multiprotocol BGP extensions for IP multicast, bidirectional forwarding detection (BFD), Enhanced Interior Gateway Routing Protocol (EIGRP), Interior Gateway Routing Protocol (IGRP), Intermediate System-to-Intermediate System (IS-IS), on-demand routing (ODR), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).
Cisco IOS IP SLAs Configuration Guide *Cisco IOS XE IP SLAs Configuration Guide* *Cisco IOS IP SLAs Command Reference*	Cisco IOS IP Service Level Agreements (IP SLAs).
Cisco IOS IP Switching Configuration Guide *Cisco IOS XE IP Switching Configuration Guide* *Cisco IOS IP Switching Command Reference*	Cisco Express Forwarding, fast switching, and Multicast Distributed Switching (MDS).
Cisco IOS IPv6 Configuration Guide *Cisco IOS XE IPv6 Configuration Guide* *Cisco IOS IPv6 Command Reference*	For IPv6 features, protocols, and technologies, go to the IPv6 “Start Here” document at the following URL: http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-roadmap.html
Cisco IOS ISO CLNS Configuration Guide *Cisco IOS XE ISO CLNS Configuration Guide* *Cisco IOS ISO CLNS Command Reference*	ISO connectionless network service (CLNS).
Cisco IOS LAN Switching Configuration Guide *Cisco IOS XE LAN Switching Configuration Guide* *Cisco IOS LAN Switching Command Reference*	VLANs, Inter-Switch Link (ISL) encapsulation, IEEE 802.10 encapsulation, IEEE 802.1Q encapsulation, and multilayer switching (MLS).
Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide *Cisco IOS Mobile Wireless Gateway GPRS Support Node Command Reference*	Cisco IOS Gateway GPRS Support Node (GGSN) in a 2.5-generation general packet radio service (GPRS) and 3-generation universal mobile telecommunication system (UMTS) network.
Cisco IOS Mobile Wireless Home Agent Configuration Guide *Cisco IOS Mobile Wireless Home Agent Command Reference*	Cisco Mobile Wireless Home Agent, an anchor point for mobile terminals for which mobile IP or proxy mobile IP services are provided.
Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide *Cisco IOS Mobile Wireless Packet Data Serving Node Command Reference*	Cisco Packet Data Serving Node (PDSN), a wireless gateway that is between the mobile infrastructure and standard IP networks and that enables packet data services in a code division multiple access (CDMA) environment.
Cisco IOS Mobile Wireless Radio Access Networking Configuration Guide *Cisco IOS Mobile Wireless Radio Access Networking Command Reference*	Cisco IOS radio access network products.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Multiprotocol Label Switching Configuration Guide *Cisco IOS XE Multiprotocol Label Switching Configuration Guide* *Cisco IOS Multiprotocol Label Switching Command Reference*	MPLS Label Distribution Protocol (LDP), MPLS Layer 2 VPNs, MPLS Layer 3 VPNs, MPLS Traffic Engineering (TE), and MPLS Embedded Management (EM) and MIBs.
Cisco IOS Multi-Topology Routing Configuration Guide *Cisco IOS Multi-Topology Routing Command Reference*	Unicast and multicast topology configurations, traffic classification, routing protocol support, and network management support.
Cisco IOS NetFlow Configuration Guide *Cisco IOS XE NetFlow Configuration Guide* *Cisco IOS NetFlow Command Reference*	Network traffic data analysis, aggregation caches, export features.
Cisco IOS Network Management Configuration Guide *Cisco IOS XE Network Management Configuration Guide* *Cisco IOS Network Management Command Reference*	Basic system management; system monitoring and logging; troubleshooting, logging, and fault management; Cisco Discovery Protocol; Cisco IOS Scripting with Tool Control Language (Tcl); Cisco networking services (CNS); DistributedDirector; Embedded Event Manager (EEM); Embedded Resource Manager (ERM); Embedded Syslog Manager (ESM); HTTP; Remote Monitoring (RMON); SNMP; and VPN Device Manager Client for Cisco IOS Software (XSM Configuration).
Cisco IOS Novell IPX Configuration Guide *Cisco IOS XE Novell IPX Configuration Guide* *Cisco IOS Novell IPX Command Reference*	Novell Internetwork Packet Exchange (IPX) protocol.
Cisco IOS Optimized Edge Routing Configuration Guide *Cisco IOS Optimized Edge Routing Command Reference*	Optimized edge routing (OER) monitoring, policy configuration, routing control, logging and reporting, and VPN IPsec/generic routing encapsulation (GRE) tunnel interface optimization.
Cisco IOS Quality of Service Solutions Configuration Guide *Cisco IOS XE Quality of Service Solutions Configuration Guide* *Cisco IOS Quality of Service Solutions Command Reference*	Class-based weighted fair queuing (CBWFQ), custom queuing, distributed traffic shaping (DTS), generic traffic shaping (GTS), IP- to-ATM class of service (CoS), low latency queuing (LLQ), modular QoS CLI (MQC), Network-Based Application Recognition (NBAR), priority queuing, Security Device Manager (SDM), Multilink PPP (MLPPP) for QoS, header compression, AutoQoS, QoS features for voice, Resource Reservation Protocol (RSVP), weighted fair queuing (WFQ), and weighted random early detection (WRED).
Cisco IOS Security Configuration Guide *Cisco IOS XE Security Configuration Guide* *Cisco IOS Security Command Reference*	Access control lists (ACLs), authentication, authorization, and accounting (AAA), firewalls, IP security and encryption, neighbor router authentication, network access security, network data encryption with router authentication, public key infrastructure (PKI), RADIUS, TACACS+, terminal access security, and traffic filters.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Service Selection Gateway Configuration Guide *Cisco IOS Service Selection Gateway Command Reference*	Subscriber authentication, service access, and accounting.
Cisco IOS Software Activation Configuration Guide *Cisco IOS Software Activation Command Reference*	An orchestrated collection of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.
Cisco IOS Software Modularity Installation and Configuration Guide *Cisco IOS Software Modularity Command Reference*	Installation and basic configuration of software modularity images, including installations on single and dual route processors, installation rollbacks, software modularity binding, software modularity processes and patches.
Cisco IOS Terminal Services Configuration Guide *Cisco IOS Terminal Services Command Reference* *Cisco IOS XE Terminal Services Command Reference*	DEC, local-area transport (LAT), and X.25 packet assembler/disassembler (PAD).
Cisco IOS Virtual Switch Command Reference	Virtual switch redundancy, high availability, and packet handling; converting between standalone and virtual switch modes; virtual switch link (VSL); Virtual Switch Link Protocol (VSLP). **Note** For information about virtual switch configuration, refer to the product-specific software configuration information for the Cisco Catalyst 6500 series switch or for the Metro Ethernet 6500 series switch.
Cisco IOS Voice Configuration Library *Cisco IOS Voice Command Reference*	Cisco IOS support for voice call control protocols, interoperability, physical and virtual interface management, and troubleshooting. The library includes documentation for IP telephony applications.
Cisco IOS VPDN Configuration Guide *Cisco IOS XE VPDN Configuration Guide* *Cisco IOS VPDN Command Reference*	Layer 2 Tunneling Protocol (L2TP) dial-out load balancing and redundancy, L2TP extended failover, L2TP security VPDN, multihop by Dialed Number Identification Service (DNIS), timer and retry enhancements for L2TP and Layer 2 Forwarding (L2F), RADIUS Attribute 82: tunnel assignment ID, shell-based authentication of VPDN users, tunnel authentication via RADIUS on tunnel terminator.
Cisco IOS Wide-Area Networking Configuration Guide *Cisco IOS XE Wide-Area Networking Configuration Guide* *Cisco IOS Wide-Area Networking Command Reference*	Frame Relay, Layer 2 Tunneling Protocol Version 3 (L2TPv3), Link Access Procedure, Balanced (LAPB), Switched Multimegabit Data Service (SMDS), and X.25.
Cisco IOS Wireless LAN Configuration Guide *Cisco IOS Wireless LAN Command Reference*	Broadcast key rotation, IEEE 802.11x support, IEEE 802.1x authenticator, IEEE 802.1x local authentication service for Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling (EAP-FAST), Multiple Basic Service Set ID (BSSID), Wi-Fi Multimedia (WMM) required elements, and Wi-Fi Protected Access (WPA).

Table 2 Cisco IOS Supplementary Documents and Resources

Document Title	Description
Cisco IOS Master Command List, All Releases	Alphabetical list of all the commands documented in all Cisco IOS releases.
Cisco IOS New, Modified, Removed, and Replaced Commands	List of all the new, modified, removed, and replaced commands for a Cisco IOS release.
Cisco IOS Software System Messages	List of Cisco IOS system messages and descriptions. System messages may indicate problems with your system; be informational only; or may help diagnose problems with communications lines, internal hardware, or the system software.
Cisco IOS Debug Command Reference	Alphabetical list of **debug** commands including brief descriptions of use, command syntax, and usage guidelines.
Release Notes and Caveats	Information about new and changed features, system requirements, and other useful information about specific software releases; information about defects in specific Cisco IOS software releases.
MIBs	Files used for network monitoring. To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator at the following URL: http://www.cisco.com/go/mibs
RFCs	Standards documents maintained by the Internet Engineering Task Force (IETF) that Cisco IOS documentation references where applicable. The full text of referenced RFCs may be obtained at the following URL: http://www.rfc-editor.org/

Additional Resources and Documentation Feedback

What's New in Cisco Product Documentation is published monthly and describes all new and revised Cisco technical documentation. The *What's New in Cisco Product Documentation* publication also provides information about obtaining the following resources:

- Technical documentation
- Cisco product security overview
- Product alerts and field notices
- Technical assistance

Cisco IOS technical documentation includes embedded feedback forms where you can rate documents and provide suggestions for improvement. Your feedback helps us improve our documentation.

Using the Command-Line Interface in Cisco IOS and Cisco IOS XE Software

Last updated: August 6, 2008

This document provides basic information about the command-line interface (CLI) in Cisco IOS and Cisco IOS XE software and how you can use some of the CLI features. This document contains the following sections:

- Initially Configuring a Device, page i
- Using the CLI, page ii
- Saving Changes to a Configuration, page xii
- Additional Information, page xii

For more information about using the CLI, see the "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information about the software documentation set, see the "About Cisco IOS and Cisco IOS XE Software Documentation" document.

Initially Configuring a Device

Initially configuring a device varies by platform. For information about performing an initial configuration, see the hardware installation documentation that is provided with the original packaging of the product or go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

After you have performed the initial configuration and connected the device to your network, you can configure the device by using the console port or a remote access method, such as Telnet or Secure Shell (SSH), to access the CLI or by using the configuration method provided on the device, such as Security Device Manager.

Changing the Default Settings for a Console or AUX Port

There are only two changes that you can make to a console port and an AUX port:

- Change the port speed with the **config-register 0x** command. Changing the port speed is not recommended. The well-known default speed is 9600.
- Change the behavior of the port; for example, by adding a password or changing the timeout value.

Note The AUX port on the Route Processor (RP) installed in a Cisco ASR1000 series router does not serve any useful customer purpose and should be accessed only under the advisement of a customer support representative.

Using the CLI

This section describes the following topics:

Understanding Command Modes

The CLI command mode structure is hierarchical, and each mode supports a set of specific commands. This section describes the most common of the many modes that exist.

Table 1 lists common command modes with associated CLI prompts, access and exit methods, and a brief description of how each mode is used.

Table 1 CLI Command Modes

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
User EXEC	Log in.	`Router>`	Issue the **logout** or **exit** command.	• Change terminal settings. • Perform basic tests. • Display device status.
Privileged EXEC	From user EXEC mode, issue the **enable** command.	`Router#`	Issue the **disable** command or the **exit** command to return to user EXEC mode.	• Issue **show** and **debug** commands. • Copy images to the device. • Reload the device. • Manage device configuration files. • Manage device file systems.
Global configuration	From privileged EXEC mode, issue the **configure terminal** command.	`Router(config)#`	Issue the **exit** command or the **end** command to return to privileged EXEC mode.	Configure the device.
Interface configuration	From global configuration mode, issue the **interface** command.	`Router(config-if)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual interfaces.
Line configuration	From global configuration mode, issue the **line vty** or **line console** command.	`Router(config-line)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual terminal lines.

Table 1 CLI Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
ROM monitor	From privileged EXEC mode, issue the **reload** command. Press the **Break** key during the first 60 seconds while the system is booting.	`rommon # >` The # symbol represents the line number and increments at each prompt.	Issue the **continue** command.	• Run as the default operating mode when a valid image cannot be loaded. • Access the fall-back procedure for loading an image when the device lacks a valid image and cannot be booted. • Perform password recovery when a CTRL-Break sequence is issued within 60 seconds of a power-on or reload event.
Diagnostic (available only on the Cisco ASR1000 series router)	The router boots or enters diagnostic mode in the following scenarios. When a Cisco IOS process or processes fail, in most scenarios the router will reload. • A user-configured access policy was configured using the **transport-map** command, which directed the user into diagnostic mode. • The router was accessed using an RP auxiliary port. • A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) was entered, and the router was configured to enter diagnostic mode when the break signal was received.	`Router(diag)#`	If a Cisco IOS process failure is the reason for entering diagnostic mode, the failure must be resolved and the router must be rebooted to exit diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or using a method that is configured to connect to the Cisco IOS CLI. If the RP auxiliary port was used to access the router, use another port for access. Accessing the router through the auxiliary port is not useful for customer purposes.	• Inspect various states on the router, including the Cisco IOS state. • Replace or roll back the configuration. • Provide methods of restarting the Cisco IOS software or other processes. • Reboot hardware, such as the entire router, an RP, an ESP, a SIP, a SPA, or possibly other hardware components. • Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

EXEC commands are not saved when the software reboots. Commands that you issue in a configuration mode can be saved to the startup configuration. If you save the running configuration to the startup configuration, these commands will execute when the software is rebooted. Global configuration mode is the highest level of configuration mode. From global configuration mode, you can enter a variety of other configuration modes, including protocol-specific modes.

ROM monitor mode is a separate mode that is used when the software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode. Use the question symbol (?) to view the commands that you can use while the device is in ROM monitor mode.

```
rommon 1 > ?
alias                set and display aliases command
boot                 boot up an external process
confreg              configuration register utility
cont                 continue executing a downloaded image
context              display the context of a loaded image
cookie               display contents of cookie PROM in hex
.
.
.
rommon 2 >
```

The following example shows how the command prompt changes to indicate a different command mode:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 1/1
Router(config-if)# ethernet
Router(config-line)# exit
Router(config)# end
Router#
```

Note A keyboard alternative to the **end** command is Ctrl-Z.

Using the Interactive Help Feature

The CLI includes an interactive Help feature. Table 2 describes how to use the Help feature.

Table 2 CLI Interactive Help Commands

Command	Purpose
help	Provides a brief description of the help feature in any command mode.
?	Lists all commands available for a particular command mode.
*partial command***?**	Provides a list of commands that begin with the character string (no space between the command and the question mark).
partial command<**Tab**>	Completes a partial command name (no space between the command and <Tab>).
command **?**	Lists the keywords, arguments, or both associated with the command (space between the command and the question mark).
command keyword **?**	Lists the arguments that are associated with the keyword (space between the keyword and the question mark).

The following examples show how to use the help commands:

help

```
Router> help
Help may be requested at any point in a command by entering a question mark '?'.  If
nothing matches, the help list will be empty and you must backup until entering a '?'
shows the available options.
Two styles of help are provided:
1. Full help is available when you are ready to enter a command argument (e.g. 'show ?')
and describes each possible argument.
2. Partial help is provided when an abbreviated argument is entered and you want to know
what arguments match the input (e.g. 'show pr?'.)
```

?

```
Router# ?
Exec commands:
  access-enable         Create a temporary access-List entry
  access-profile        Apply user-profile to interface
  access-template       Create a temporary access-List entry
  alps                  ALPS exec commands
  archive               manage archive files
<snip>
```

partial command?

```
Router(config)# zo?
zone  zone-pair
```

partial command<Tab>

```
Router(config)# we<Tab> webvpn
```

command ?

```
Router(config-if)# pppoe ?
  enable        Enable pppoe
  max-sessions  Maximum PPPOE sessions
```

command keyword ?

```
Router(config-if)# pppoe enable ?
  group  attach a BBA group
  <cr>
```

Understanding Command Syntax

Command syntax is the format in which a command should be entered in the CLI. Commands include the name of the command, keywords, and arguments. Keywords are alphanumeric strings that are used literally. Arguments are placeholders for values that a user must supply. Keywords and arguments may be required or optional.

Specific conventions convey information about syntax and command elements. Table 3 describes these conventions.

Table 3 CLI Syntax Conventions

Symbol/Text	Function	Notes
< > (angle brackets)	Indicate that the option is an argument.	Sometimes arguments are displayed without angle brackets.
A.B.C.D.	Indicates that you must enter a dotted decimal IP address.	Angle brackets (< >) are not always used to indicate that an IP address is an argument.
WORD (all capital letters)	Indicates that you must enter one word.	Angle brackets (< >) are not always used to indicate that a WORD is an argument.
LINE (all capital letters)	Indicates that you must enter more than one word.	Angle brackets (< >) are not always used to indicate that a LINE is an argument.
<cr> (carriage return)	Indicates the end of the list of available keywords and arguments, and also indicates when keywords and arguments are optional. When <cr> is the only option, you have reached the end of the branch or the end of the command if the command has only one branch.	—

The following examples show syntax conventions:

```
Router(config)# ethernet cfm domain ?
  WORD  domain name
Router(config)# ethernet cfm domain dname ?
  level
Router(config)# ethernet cfm domain dname level ?
  <0-7>  maintenance level number
Router(config)# ethernet cfm domain dname level 7 ?
  <cr>
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
Router(config)# logging host ?
  Hostname or A.B.C.D  IP address of the syslog server
  ipv6                 Configure IPv6 syslog server
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
```

Understanding Enable and Enable Secret Passwords

Some privileged EXEC commands are used for actions that impact the system, and it is recommended that you set a password for these commands to prevent unauthorized use. Two types of passwords, enable (not encrypted) and enable secret (encrypted), can be set. The following commands set these passwords and are issued in global configuration mode:

- **enable** *password*
- **enable secret** *password*

Using an enable secret password is recommended because it is encrypted and more secure than the enable password. When you use an enable secret password, text is encrypted (unreadable) before it is written to the config.text file. When you use an enable password, the text is written as entered (readable) to the config.text file.

Each type of password is case sensitive, can contain from 1 to 25 uppercase and lowercase alphanumeric characters, and can start with a number. Spaces are also valid password characters; for example, “two words” is a valid password. Leading spaces are ignored, but trailing spaces are recognized.

Note Both password commands have numeric keywords that are single integer values. If you choose a number for the first character of your password followed by a space, the system will read the number as if it were the numeric keyword and not as part of your password.

When both passwords are set, the enable secret password takes precedence over the enable password.

To remove a password, use the **no** form of the commands: **no enable** *password* or **no enable secret** *password*.

For more information about password recovery procedures for Cisco products, see http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_tech_note09186a00801746e6.shtml.

Using the Command History Feature

The CLI command history feature saves the commands you enter during a session in a command history buffer. The default number of commands saved is 10, but the number is configurable within the range of 0 to 256. This command history feature is particularly useful for recalling long or complex commands.

To change the number of commands saved in the history buffer for a terminal session, issue the **terminal history size** command:

```
Router# terminal history size num
```

A command history buffer is also available in line configuration mode with the same default and configuration options. To set the command history buffer size for a terminal session in line configuration mode, issue the **history** command:

```
Router(config-line)# history [size num]
```

To recall commands from the history buffer, use the following methods:

- Press Ctrl-P or the up arrow key—Recalls commands beginning with the most recent command. Repeat the key sequence to recall successively older commands.

- Press Ctrl-N or the down arrow key—Recalls the most recent commands in the history buffer after they have been recalled using Ctrl-P or the up arrow key. Repeat the key sequence to recall successively more recent commands.

Note The arrow keys function only on ANSI-compatible terminals such as the VT100.

- Issue the **show history** command in user EXEC or privileged EXEC mode—Lists the most recent commands that you entered. The number of commands that are displayed is determined by the setting of the **terminal history size** and **history** commands.

  The CLI command history feature is enabled by default. To disable this feature for a terminal session, issue the **terminal no history** command in user EXEC or privileged EXEC mode or the **no history** command in line configuration mode.

Abbreviating Commands

Typing a complete command name is not always required for the command to execute. The CLI recognizes an abbreviated command when the abbreviation contains enough characters to uniquely identify the command. For example, the **show version** command can be abbreviated as **sh ver**. It cannot be abbreviated as **s ver** because **s** could mean **show**, **set**, or **systat**. The **sh v** abbreviation also is not valid because the **show** command has **vrrp** as a keyword in addition to **version**. (Command and keyword examples from Cisco IOS Release 12.4(13)T.)

Using Aliases for CLI Commands

To save time and the repetition of entering the same command multiple times, you can use a command alias. An alias can be configured to do anything that can be done at the command line, but an alias cannot move between modes, type in passwords, or perform any interactive functions.

Table 4 shows the default command aliases.

Table 4 Default Command Aliases

Command Alias	Original Command
h	help
lo	logout
p	ping
s	show
u or **un**	undebug
w	where

To create a command alias, issue the **alias** command in global configuration mode. The syntax of the command is **alias** *mode command-alias original-command*. Following are some examples:

- Router(config)# **alias exec prt partition**—privileged EXEC mode
- Router(config)# **alias configure sb source-bridge**—global configuration mode
- Router(config)# **alias interface rl rate-limit**—interface configuration mode

To view both default and user-created aliases, issue the **show alias** command.

For more information about the **alias** command, see http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html.

Using the no and default Forms of Commands

Most configuration commands have a **no** form that is used to reset a command to its default value or disable a feature or function. For example, the **ip routing** command is enabled by default. To disable this command, you would issue the **no ip routing** command. To re-enable IP routing, you would issue the **ip routing** command.

Configuration commands may also have a **default** form, which returns the command settings to their default values. For commands that are disabled by default, using the **default** form has the same effect as using the **no** form of the command. For commands that are enabled by default and have default settings, the **default** form enables the command and returns the settings to their default values.

The **no** and **default** forms of commands are described in the command pages of command references.

Using the debug Command

A **debug** command produces extensive output that helps you troubleshoot problems in your network. These commands are available for many features and functions within Cisco IOS and Cisco IOS XE software. Some **debug** commands are **debug all**, **debug aaa accounting**, and **debug mpls packets**. To use **debug** commands during a Telnet session with a device, you must first enter the **terminal monitor** command. To turn off debugging completely, you must enter the **undebug all** command.

For more information about **debug** commands, see the *Cisco IOS Debug Command Reference* at http://www.cisco.com/en/US/docs/ios/debug/command/reference/db_book.html.

Caution Debugging is a high priority and high CPU utilization process that can render your device unusable. Use **debug** commands only to troubleshoot specific problems. The best times to run debugging are during periods of low network traffic and when few users are interacting with the network. Debugging during these periods decreases the likelihood that the **debug** command processing overhead will affect network performance or user access or response times.

Filtering Output Using Output Modifiers

Many commands produce lengthy output that may use several screens to display. Using output modifiers, you can filter this output to show only the information that you want to see.

Three output modifiers are available and are described as follows:

- **begin** *regular expression*—Displays the first line in which a match of the regular expression is found and all lines that follow.
- **include** *regular expression*—Displays all lines in which a match of the regular expression is found.
- **exclude** *regular expression*—Displays all lines except those in which a match of the regular expression is found.

To use one of these output modifiers, type the command followed by the pipe symbol (|), the modifier, and the regular expression that you want to search for or filter. A regular expression is a case-sensitive alphanumeric pattern. It can be a single character or number, a phrase, or a more complex string.

The following example illustrates how to filter output of the **show interface** command to display only lines that include the expression "protocol."

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

Understanding CLI Error Messages

You may encounter some error messages while using the CLI. Table 5 shows the common CLI error messages.

Table 5 Common CLI Error Messages

Error Message	Meaning	How to Get Help
% Ambiguous command: "show con"	You did not enter enough characters for the command to be recognized.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Incomplete command.	You did not enter all the keywords or values required by the command.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Invalid input detected at "^" marker.	You entered the command incorrectly. The caret (^) marks the point of the error.	Enter a question mark (?) to display all the commands that are available in this command mode. The keywords that you are allowed to enter for the command appear.

For more system error messages, see the following documents:

- *Cisco IOS Release 12.2SR System Message Guide*
- *Cisco IOS System Messages, Volume 1 of 2* (Cisco IOS Release 12.4)
- *Cisco IOS System Messages, Volume 2 of 2* (Cisco IOS Release 12.4)

Saving Changes to a Configuration

To save changes that you made to the configuration of a device, you must issue the **copy running-config startup-config** command or the **copy system:running-config nvram:startup-config** command. When you issue these commands, the configuration changes that you made are saved to the startup configuration and saved when the software reloads or power to the device is turned off or interrupted. The following example shows the syntax of the **copy running-config startup-config** command:

```
Router# copy running-config startup-config
Destination filename [startup-config]?
```

You press Enter to accept the startup-config filename (the default), or type a new filename and then press Enter to accept that name. The following output is displayed indicating that the configuration was saved:

```
Building configuration...
[OK]
Router#
```

On most platforms, the configuration is saved to NVRAM. On platforms with a Class A flash file system, the configuration is saved to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Additional Information

- "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*:

  http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/cf_cli-basics.html

  or

  "Using Cisco IOS XE Software" chapter of the *Cisco ASR1000 Series Aggregation Services Routers Software Configuration Guide*:

  http://www.cisco.com/en/US/docs/routers/asr1000/configuration/guide/chassis/using_cli.html

- Cisco Product Support Resources

  http://www.cisco.com/web/psa/products/index.html

- Support area on Cisco.com (also search for documentation by task or product)

  http://www.cisco.com/en/US/support/index.html

- *White Paper: Cisco IOS Reference Guide*

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1828/products_white_paper09186a008018305e.shtml

- Software Download Center (downloads; tools; licensing, registration, advisory, and general information) (requires Cisco.com User ID and password)

  http://www.cisco.com/kobayashi/sw-center/

- Error Message Decoder, a tool to help you research and resolve error messages for Cisco IOS software

  http://www.cisco.com/pcgi-bin/Support/Errordecoder/index.cgi

- Command Lookup Tool, a tool to help you find detailed descriptions of Cisco IOS commands (requires Cisco.com user ID and password)

  http://tools.cisco.com/Support/CLILookup

- Output Interpreter, a troubleshooting tool that analyzes command output of supported **show** commands

  https://www.cisco.com/pcgi-bin/Support/OutputInterpreter/home.pl\

Using the Cisco IOS Command-Line Interface (CLI)

Using the Cisco IOS Command-Line Interface

Last Updated: May 2, 2008

The Cisco IOS command-line interface (CLI) is the primary user interface used for configuring, monitoring, and maintaining Cisco devices. This user interface allows you to directly and simply execute Cisco IOS commands, whether using a router console or terminal, or using remote access methods.

This chapter describes the basic features of the Cisco IOS CLI and how to use them. Topics covered include an introduction to Cisco IOS command modes, navigation and editing features, help features, and command history features.

Additional user interfaces include Setup mode (used for first-time startup), the Cisco Web Browser, and user menus configured by a system administrator. For information about Setup mode, see Using Setup Mode to Configure a Cisco Networking Device and Using AutoInstall to Remotely Configure Cisco NetworkingDevices. For information on issuing commands using the Cisco Web Browser, see Using the Cisco Web Browser User Interface. For information on user menus, see Managing Connections, Menus, and System Banners.

For a complete description of the user interface commands in this chapter, see the *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Master Command List, All Releases*.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Cisco IOS CLI Command Modes Overview

To aid in the configuration of Cisco devices, the Cisco IOS command-line interface is divided into different command modes. Each command mode has its own set of commands available for the configuration, maintenance, and monitoring of router and network operations. The commands available to you at any given time depend on the mode you are in. Entering a question mark (**?**) at the system prompt (router prompt) allows you to obtain a list of commands available for each command mode.

The use of specific commands allows you to navigate from one command mode to another. The standard order that a user would access the modes is as follows: user EXEC mode; privileged EXEC mode; global configuration mode; specific configuration modes; configuration submodes; and configuration subsubmodes.

When you start a session on a router, you generally begin in *user EXEC mode*, which is one of two access levels of the EXEC mode. For security purposes, only a limited subset of Exec commands are available in user EXEC mode. This level of access is reserved for tasks that do not change the configuration of the router, such as determining the router status.

In order to have access to all commands, you must enter *privileged EXEC mode*, which is the second level of access for the EXEC mode. Normally, you must enter a password to enter privileged EXEC mode. In privileged EXEC mode, you can enter any EXEC command, because privileged EXEC mode is a superset of the user EXEC mode commands.

Most EXEC mode commands are one-time commands, such as **show** or **more** commands, which show the current configuration status, and **clear** commands, which clear counters or interfaces. EXEC mode commands are not saved across reboots of the router.

From privileged EXEC mode, you can enter *global configuration mode*. In this mode, you can enter commands that configure general system characteristics. You also can use global configuration mode to enter specific configuration modes. Configuration modes, including global configuration mode, allow you to make changes to the running configuration. If you later save the configuration, these commands are stored across router reboots.

From global configuration mode you can enter a variety of protocol-specific or feature-specific configuration modes. The CLI hierarchy requires that you enter these specific configuration modes only through global configuration mode. As an example, this chapter describes *interface configuration mode*, a commonly used configuration mode.

From configuration modes, you can enter configuration submodes. Configuration submodes are used for the configuration of specific features within the scope of a given configuration mode. As an example, this chapter describes the *subinterface configuration mode*, a submode of the interface configuration mode.

ROM monitor mode is a separate mode used when the router cannot boot properly. If your system (router, switch, or access server) does not find a valid system image to load when it is booting, the system will enter ROM monitor mode. ROM monitor (ROMMON) mode can also be accessed by interrupting the boot sequence during startup.

The following sections contain detailed information on these command modes:

- User EXEC mode, page 3
- Privileged EXEC Mode, page 4
- Global Configuration Mode, page 5
- Interface Configuration Mode, page 6
- Subinterface Configuration Mode, page 7
- ROM Monitor Mode, page 8

Table 1 follows these sections and summarizes the main Cisco IOS command modes.

User EXEC mode

Logging in to the router places you in user EXEC command mode (unless the system is configured to take you immediately to privileged EXEC mode). Typically, login will require a username and a password. You may try three times to enter a password before the connection attempt is refused.

Note For information on setting the password, see Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices.

The Exec commands available at the user level are a subset of those available at the privileged level. In general, the user EXEC commands allow you to connect to remote devices, change terminal line settings on a temporary basis, perform basic tests, and list system information.

To list the available user EXEC commands, use the following command:

Command	Purpose
`Router> ?`	Lists the user EXEC commands.

The user EXEC mode prompt consists of the hostname of the device followed by an angle bracket (>), as shown in the following example:

```
Router>
```

The default host name is generally `Router,` unless it has been changed during initial configuration using the **setup** Exec command. You also change the hostname using the **hostname** global configuration command.

Note Examples in Cisco IOS documentation assume the use of the default name of "Router." Different devices (for example, access servers) may use a different default name. If the routing device (router, access server, or switch) has been named with the **hostname** command, that name will appear as the prompt instead of the default name.

To list the commands available in user EXEC mode, enter a question mark (**?**) as shown in the following example:

```
Router> ?

Exec commands:
 <1-99>           Session number to resume
 connect          Open a terminal connection
 disconnect       Disconnect an existing telnet session
 enable           Turn on privileged commands
 exit             Exit from Exec mode
 help             Description of the interactive help system
 lat              Open a lat connection
 lock             Lock the terminal
 login            Log in as a particular user
 logout           Exit from Exec mode and log out
 menu             Start a menu-based user interface
 mbranch          Trace multicast route for branch of tree
 mrbranch         Trace reverse multicast route to branch of tree
```

```
mtrace           Trace multicast route to group
name-connection  Name an existing telnet connection
pad              Open a X.29 PAD connection
ping             Send echo messages
resume           Resume an active telnet connection
show             Show running system information
systat           Display information about terminal lines
telnet           Open a telnet connection
terminal         Set terminal line parameters
tn3270           Open a tn3270 connection
trace            Trace route to destination
where            List active telnet connections
x3               Set X.3 parameters on PAD
```

The list of commands will vary depending on the software feature set and router platform you are using.

Note You can enter commands in uppercase, lowercase, or mixed case. Only passwords are case sensitive. However, Cisco IOS documentation convention is to always present commands in lowercase.

Privileged EXEC Mode

Because many privileged EXEC mode commands set operating parameters, privileged-level access should be password protected to prevent unauthorized use. The privileged EXEC command set includes those commands contained in user EXEC mode. Privileged EXEC mode also provides access to configuration modes through the **configure** command, and includes advanced testing commands, such as **debug**.

The privileged EXEC mode prompt consists of the hostname of the device followed by a pound sign (#), as shown in the following example:

```
Router#
```

To access privileged EXEC mode, use the following command:

Command	Purpose
`Router>` **`enable`**	Enables privileged EXEC mode. • Enter your password when prompted.

Note that privileged EXEC mode is sometimes referred to as “enable mode,” because the **enable** command is used to enter the mode.

If a password has been configured on the system, you will be prompted to enter it before being allowed access to privileged EXEC mode. The password is not displayed on the screen and is case sensitive. If an enable password has not been set, privileged EXEC mode can be accessed only from the router console (terminal connected to the console port). The system administrator uses the **enable secret** or **enable password** global configuration command to set the password that restricts access to privileged mode. For information on setting the passwords, see the “Configuring Passwords and Privileges” chapter in the Release 12.4 *Cisco IOS Security Configuration Guide.*

To return to user EXEC mode, use the following command:

Command	Purpose
Router# **disable**	Exits from privileged EXEC mode to user EXEC mode.

The following example shows the process of accessing privileged EXEC mode:

```
Router> enable
Password:<letmein>
Router#
```

Note that the password will not be displayed as you type, but is shown here for illustrational purposes. To list the commands available in privileged EXEC mode, issue the **?** command at the prompt. From privileged EXEC mode you can access global configuration mode, which is described in the following section.

Note Because the privileged EXEC command set contains all of the commands available in user EXEC mode, some commands can be entered in either mode. In Cisco IOS documentation, commands that can be entered in either user EXEC mode or privileged EXEC mode are referred to as EXEC mode commands. If user or privileged EXEC mode is not specified in the documentation, assume that you can enter the referenced commands in either mode.

Global Configuration Mode

The term “global” is used to indicate characteristics or features that affect the system as a whole. Global configuration mode is used to configure your system globally, or to enter specific configuration modes to configure specific elements such as interfaces or protocols. Use the **configure terminal** privileged EXEC mode command to enter global configuration mode.

To access global configuration mode, use the following command in privileged EXEC mode:

Command	Purpose
Router# **configure terminal**	From privileged EXEC mode, enters global configuration mode.

The following example shows the process of entering global configuration mode from privileged EXEC mode:

```
Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Note that the system prompt changes to indicate that you are now in global configuration mode. The prompt for global configuration mode consists of the hostname of the device followed by `(config)` and the pound sign (`#`). To list the commands available in privileged EXEC mode, issue the **?** command at the prompt.

Commands entered in global configuration mode update the running configuration file as soon as they are entered. In other words, changes to the configuration take effect each time you press the Enter or Return key at the end of a valid command. However, these changes are not saved into the startup configuration file until you issue the **copy running-config startup-config** EXEC mode command. This behavior is explained in more detail later in this document.

As shown in the example, the system dialog prompts you to end your configuration session (exit configuration mode) by pressing the Control (Ctrl) and "z" keys simultaneously; when you press these keys, **^Z** appears on screen. You can actually end your configuration session by entering the Ctrl-Z key combination, using the **end** command, or using the Ctrl-C key combination. The **end** command is the recommended way to indicate to the system that you are done with the current configuration session.

Note If you use Ctrl-Z at the end of a command line in which a valid command has been typed, that command will be added to the running configuration file. In other words, using Ctrl-Z is equivalent to hitting the Enter (Carriage Return) key before exiting. For this reason, it is safer to end your configuration session using the **end** command. Alternatively, you can use the Ctrl-C key combination to end your configuration session without sending a Carriage Return signal.

You can also use the **exit** command to return from global configuration mode to EXEC mode, but this works only in global configuration mode. Pressing Ctrl-Z or entering the **end** command will always take you back to EXEC mode regardless of which configuration mode or configuration submode you are in.

To exit global configuration command mode and return to privileged EXEC mode, use one of the following commands:

Command	Purpose
`Router(config)# end` or `Router(config)# ^Z`	Ends the current configuration session and returns to privileged EXEC mode.
`Router(config)# exit`	Exits the current command mode and returns to the preceding mode. For example, exits from global configuration mode to privileged EXEC mode.

From global configuration mode, you can enter a number of protocol-specific, platform-specific, and feature-specific configuration modes. Information about specific modes is given in task-specific contexts throughout the Cisco IOS software documentation set.

Interface configuration mode, described in the following section, is an example of a configuration mode you can enter from global configuration mode.

Interface Configuration Mode

One example of a specific configuration mode you enter from global configuration mode is interface configuration mode.

Many features are enabled on a per-interface basis. Interface configuration commands modify the operation of an interface such as an Ethernet, FDDI, or serial port. Interface configuration commands always follow an **interface** global configuration command, which defines the interface type.

For details on interface configuration commands that affect general interface parameters, such as bandwidth or clock rate, refer to the *Cisco IOS Interface and Hardware Component Configuration Guide* for your release. For protocol-specific commands, refer to the appropriate Cisco IOS software command reference.

To access and list the interface configuration commands, use the following command:

Command	Purpose
`Router(config)# interface type number`	Specifies the interface to be configured, and enters interface configuration mode.

In the following example, the user enter interface configuration mode for serial interface 0. The new prompt, `hostname(config-if)#`, indicates interface configuration mode.

```
Router(config)# interface serial 0
Router(config-if)#
```

To exit interface configuration mode and return to global configuration mode, enter the **exit** command.

Configuration submodes are configuration modes entered from other configuration modes (besides global configuration mode). Configuration submodes are for the configuration of specific elements within the configuration mode. One example of a configuration submode is subinterface configuration mode, described in the following section.

Subinterface Configuration Mode

From interface configuration mode, you can enter subinterface configuration mode. Subinterface configuration mode is a submode of interface configuration mode. In subinterface configuration mode you can configure multiple virtual interfaces (called subinterfaces) on a single physical interface. Subinterfaces appear to be distinct physical interfaces to the various protocols. For example, Frame Relay networks provide multiple point-to-point links called permanent virtual circuits (PVCs). PVCs can be grouped under separate subinterfaces that in turn are configured on a single physical interface. From a bridging spanning-tree viewpoint, each subinterface is a separate bridge port, and a frame arriving on one subinterface can be sent out on another subinterface.

Subinterfaces also allow multiple encapsulations for a protocol on a single interface. For example, a router or access server can receive an Advanced Research Projects Agency (ARPA-framed) Internetwork Packet Exchange (IPX) packet and forward the packet back out the same physical interface as a Subnetwork Access Protocol (SNAP-framed) IPX packet.

For detailed information on how to configure subinterfaces, refer to the appropriate documentation module for a specific protocol in the Cisco IOS software documentation set.

To access subinterface configuration mode, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# interface type number`	Specifies the virtual interface to be configured and enters subinterface configuration mode.

In the following example, a subinterface is configured for serial line 2, which is configured for Frame Relay encapsulation. The subinterface is identified as "2.1" to indicate that it is subinterface 1 of serial interface 2. The new prompt `hostname(config-subif)#` indicates subinterface configuration mode. The subinterface can be configured to support one or more Frame Relay PVCs.

```
Router(config)# interface serial 2
Router(config-if)# encapsulation frame-relay
Router(config-if)# interface serial 2.1
Router(config-subif)#
```

To exit subinterface configuration mode and return to interface configuration mode, use the **exit** command. To end your configuration session and return to privileged EXEC mode, press Ctrl-Z or enter the **end** command.

ROM Monitor Mode

ROM monitor mode (ROMMON) runs from a specialized software image, and is used to manually locate a valid system software image from which to boot the system (ROM monitor mode is also sometimes called "boot mode").

If your system (router, switch, or access server) does not find a valid system image to load, the system will enter ROM monitor mode. ROM monitor mode can also be accessed by interrupting the boot sequence during startup. From ROM monitor mode, you can boot the device or perform diagnostic tests.

On most systems you can enter ROM monitor mode by entering the **reload** Exec command and then issuing the Break command during the first 60 seconds of startup. The Break command is issued by pressing the Break key on your keyboard or by using the Break key-combination (the default Break key combination is Ctrl-C).

Note You must have a console connection to the router to perform this procedure, because Telnet connections will be lost when the system reboots.

To access ROM monitor mode from EXEC mode, perform the following steps:

Step 1 Enter the **reload** command in EXEC mode. After you enter this command and responding to the system prompts as necessary, the system will begin reloading the system software image.

Step 2 Issue the Break command during the first 60 seconds of system startup. The break command is issued using the Break key or Break key combination. (The default Break key combination is Ctrl-C, but this may be configured differently on your system.) Issuing the break command interrups the boot sequence and brings you into ROM monitor mode.

Another method for entering ROM monitor mode is to set the configuration register so that the router automatically enters ROM monitor mode when it boots. For information about setting the configuration register value, see "Rebooting and Reloading - Configuring Image Loading Characteristics. "

ROM monitor mode uses an angle bracket (>) as the command line prompt. On some Cisco devices the default ROM monitor prompt is `rommon >`. A list of ROM monitor commands is displayed when you enter the **?** command or **help** command. The following example shows how this list of commands may appear:

```
User break detected at location 0x8162ac6\Œ
rommon 1 > ?

alias                set and display aliases command
boot                 boot up an external process
break                set/show/clear the breakpoint
confreg              configuration register utility
cont                 continue executing a downloaded image
context              display the context of a loaded image
```

```
cpu_card_type       display CPU card type
dev                 list the device table
dir                 list files in file system
dis                 disassemble instruction stream
frame               print out a selected stack frame
help                monitor builtin command help
history             monitor command history
meminfo             main memory information
repeat              repeat a monitor command
reset               system reset
set                 show all monitor variables
stack               produce a stack trace
sync                write monitor environment to NVRAM
sysret              print out info from last system return
unalias             unset an alias
unset               unset a monitor variable
rommon 2>
```

The list of available commands will vary depending on the software image and platform you are using. Some versions of ROMMON will display a list of commands in a pre-aliased format such as the following:

```
> ?

$ state      Toggle cache state (? for help)
B [filename] [TFTP Server IP address | TFTP Server Name]
             Load and execute system image from ROM or from TFTP server
C [address]  Continue execution [optional address]
D /S M L V   Deposit value V of size S into location L with modifier M
E /S M L     Examine location L with size S with modifier M
G [address]  Begin execution
H            Help for commands
I            Initialize
K            Stack trace
L [filename] [TFTP Server IP address | TFTP Server Name]
             Load system image from ROM or from TFTP server, but do not
             begin execution
O            Show configuration register option settings
P            Set the break point
S            Single step next instruction
T function   Test device (? for help)
Deposit and Examine sizes may be B (byte), L (long) or S (short).
Modifiers may be R (register) or S (byte swap).
Register names are: D0-D7, A0-A6, SS, US, SR, and PC
```

To exit ROM monitor mode, use the **continue** command; this will restart the booting process.

For more information on ROM monitor mode characteristics and using ROM monitor mode, see the "Rebooting and Reloading - Configuring Image Loading Characteristics".

Summary of Main Cisco IOS Command Modes

Table 1 summarizes the main command modes used in the Cisco IOS CLI.

Table 1 Summary of the Main Cisco IOS Command Modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Use the **logout** command.
Privileged Exec	From user EXEC mode, use the **enable** Exec command.	`Router#`	To exit to user EXEC mode, use the **disable** command. To enter global configuration mode, use the **configure terminal** privileged EXEC command.
Global configuration	From privileged EXEC mode, use the **configure terminal** command.	`Router(config)#`	To exit to privileged EXEC mode, use the **end** command or press **Ctrl-Z**. To enter interface configuration mode, use the **interface** configuration command.
Interface configuration	From global configuration mode, enter by specifying an interface with an **interface** command.	`Router(config-if)#`	To exit to global configuration mode, use the **exit** command. To exit to privileged EXEC mode, use the **end** command or press **Ctrl-Z**. To enter subinterface configuration mode, specify a subinterface with the **interface** command.
Subinterface configuration	From interface configuration mode, specify a subinterface with an **interface** command. (The availability of this mode is dependent on your platform.)	`Router(config-subif)#`	To exit to global configuration mode, use the **exit** command. To exit to privileged EXEC mode, use the **end** command or press **Ctrl-Z**.
ROM monitor	From privileged EXEC mode, use the **reload** Exec command. Press the Break key during the first 60 seconds while the system is booting.	`>` or `boot>` or `rommon >`	If you entered ROM monitor mode by interrupting the loading process, you can exit ROM monitor mode and resume loading by using the **continue** commands.

Cisco IOS CLI Task List

To familiarize yourself with the features of the Cisco IOS CLI, perform any of the tasks described in the following sections:

- Getting Context-Sensitive Help, page 11
- Using the no and default Forms of Commands, page 15
- Using Command History, page 15
- Using CLI Editing Features and Shortcuts, page 16
- Searching and Filtering CLI Output, page 21

Getting Context-Sensitive Help

Entering a question mark (**?**) at the system prompt displays a list of commands available for each command mode. You also can get a list of the arguments and keywords available for any command with the context-sensitive help feature.

To get help specific to a command mode, a command name, a keyword, or an argument, use any of the following commands:

Command	Purpose
`(`*prompt*`)#` **help**	Displays a brief description of the help system.
`(`*prompt*`)#` *abbreviated-command-entry***?**	Lists commands in the current mode that begin with a particular character string.
`(`*prompt*`)#` *abbreviated-command-entry*<**Tab**>	Completes a partial command name.
`(`*prompt*`)#` **?**	Lists all commands available in the command mode.
`(`*prompt*`)#` *command* **?**	Lists the available syntax options (arguments and keywords) for the command.
`(`*prompt*`)#` *command keyword* **?**	Lists the next available syntax option for the command.

Note that the system prompt will vary depending on which configuration mode you are in.

When context-sensitive help is used, the space (or lack of a space) before the question mark (**?**) is significant. To obtain a list of commands that begin with a particular character sequence, type in those characters followed immediately by the question mark (**?**). Do not include a space. This form of help is called *word help,* because it completes a word for you. For more information, see the "Completing a Partial Command Name" section later in this chapter.

To list keywords or arguments, enter a question mark (**?**) in place of a keyword or argument. Include a space before the **?**. This form of help is called *command syntax help,* because it shows you which keywords or arguments are available based on the command, keywords, and arguments you already have entered.

You can abbreviate commands and keywords to the number of characters that allow a unique abbreviation. For example, you can abbreviate the **configure terminal** command to **config t**. Because the abbreviated form of the command is unique, the router will accept the abbreviated form and execute the command.

Entering the **help** command (available in any command mode) will provide the following description of the help system:

```
Router# help

Help may be requested at any point in a command by entering
a question mark '?'. If nothing matches, the help list will
be empty and you must back up until entering a '?' shows the
available options.
Two styles of help are provided:
1. Full help is available when you are ready to enter a
   command argument (e.g. 'show ?') and describes each possible
   argument.
2. Partial help is provided when an abbreviated argument is entered
   and you want to know what arguments match the input
   (e.g. 'show pr?'.)
```

As described in the **help** command output, you can use the question mark (**?**) to complete a partial command name (partial help), or to obtain a list of arguments or keywords that will complete the current command.

The following example illustrates how the context-sensitive help feature enables you to create an access list from configuration mode.

Enter the letters **co** at the system prompt followed by a question mark (**?**). Do not leave a space between the last letter and the question mark. The system provides the commands that begin with **co**.

```
Router# co?
configure  connect  copy
```

Enter the **configure** command followed by a space and a question mark to list the keywords for the command and a brief explanation:

```
Router# configure ?
  memory    Configure from NV memory
  network   Configure from a TFTP network host
  overwrite-network  Overwrite NV memory from TFTP network host
  terminal  Configure from the terminal
  <cr>
```

The `<cr>` symbol ("cr" stands for carriage return) appears in the list to indicate that one of your options is to press the Return or Enter key to execute the command, without adding any keywords. In this example, the output indicates that your options for the configure command are **configure memory** (configure from NVRAM), **configure network** (configure from a file on the network), **configure overwrite-network** (configure from a file on the network and replace the file in NVRAM), or **configure terminal** (configure manually from the terminal connection). For most commands, the <cr> symbol is used to indicate that you can execute the command with the syntax you have already entered. However, the configure command is a special case, because the CLI will prompt you for the missing syntax:

```
Router# configure
Configuring from terminal, memory, or network [terminal]? terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

The default response for the ? prompt is indicated in the CLI output by a bracketed option at the end of the line. In the preceding example, pressing the Enter (or Return) key is equivalent to typing in the word "terminal."

Enter the **configure terminal** command to enter global configuration mode:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

The CLI provides error isolation in the form of an error indicator, a caret symbol (^). The ^ symbol appears at the point in the command string where the user has entered incorrect or unrecognized command syntax. For example, the caret symbol in the following output shows the letter that was mistyped in the command:

```
Router# configure terminal
                     ^
% Invalid input detected at '^' marker.

Router#
```

Note that an error message (indicated by the % symbol) appears on the screen to alert you to the error marker.

Enter the **access-list** command followed by a space and a question mark to list the available options for the command:

```
Router(config)#access-list ?
 <1-99>            IP standard access list
 <100-199>         IP extended access list
 <1100-1199>       Extended 48-bit MAC address access list
 <1300-1999>       IP standard access list (expanded range)
 <200-299>         Protocol type-code access list
 <2000-2699>       IP extended access list (expanded range)
 <700-799>         48-bit MAC address access list
 dynamic-extended  Extend the dynamic ACL absolute timer
 rate-limit        Simple rate-limit specific access list
```

The two numbers within the angle brackets represent an inclusive range. Enter the access list number **99** and then enter another question mark to see the arguments that apply to the keyword and brief explanations:

```
Router(config)# access-list 99 ?
  deny    Specify packets to reject
  permit  Specify packets to forward
```

Enter the **deny** argument followed by a question mark (**?**) to list additional options:

```
Router(config)# access-list 99 deny ?
  A.B.C.D  Address to match
```

Generally, uppercase letters represent variables (arguments). Enter the IP address followed by a question mark (**?**) to list additional options:

```
Router(config)# access-list 99 deny 172.31.134.0 ?
  A.B.C.D  Mask of bits to ignore
  <cr>
```

In this output, A.B.C.D indicates that use of a wildcard mask is allowed. The wildcard mask is a method for matching IP addresses or ranges of IP addresses. For example, a wildcard mask of 0.0.0.255 matches any number in the range from 0 to 255 that appears in the fourth octet of an IP address.

Enter the wildcard mask followed by a question mark (**?**) to list further options:

```
Router(config)# access-list 99 deny 172.31.134.0 0.0.0.255 ?
<cr>
```

The `<cr>` symbol by itself indicates there are no more keywords or arguments. Press Enter (or Return) to execute the command.:

```
Router(config)# access-list 99 deny 172.31.134.0 0.0.0.255
```

The system adds an entry to access list 99 that denies access to all hosts on subnet 172.31.134.0, while ignoring bits for IP addresses that end in 0 to 255.

Displaying All User Exec Commands

To configure the current session to display the full set of user EXEC commands, use the following command in user EXEC or privileged EXEC mode:

Command	Purpose
`Router# terminal full-help`	Configures this session to provide help for the full set of user-level commands.

The system administrator can also configure the system to always display full help for connections made to a particular line using the **full-help** line configuration command.

The **full-help** and **terminal full-help** commands enable the displaying of all help messages available in user EXEC mode when the **show ?** command is executed.

The following example is output for the **show ?** command with the **terminal full-help command** disabled and then enabled:

```
Router> terminal no full-help
Router> show ?

  bootflash  Boot Flash information
  calendar   Display the hardware calendar
  clock      Display the system clock
  context    Show context information
  dialer     Dialer parameters and statistics
  history    Display the session command history
  hosts      IP domain-name, lookup style, nameservers, and host table
  isdn       ISDN information
  kerberos   Show Kerberos Values
  modemcap   Show Modem Capabilities database
  ppp        PPP parameters and statistics
  rmon       rmon statistics
  sessions   Information about Telnet connections
  snmp       snmp statistics
  terminal   Display terminal configuration parameters
  users      Display information about terminal lines
  version    System hardware and software status

Router> terminal full-help
Router> show ?

  access-expression  List access expression
  access-lists       List access lists
  aliases            Display alias commands
  apollo             Apollo network information
  appletalk          AppleTalk information
  arp                ARP table
  async              Information on terminal lines used as router interfaces
  bootflash          Boot Flash information
  bridge             Bridge Forwarding/Filtering Database [verbose]
  bsc                BSC interface information
  bstun              BSTUN interface information
  buffers            Buffer pool statistics
  calendar           Display the hardware calendar
  cdp                CDP information
  clns               CLNS network information
  clock              Display the system clock
  cls                DLC user information
  cmns               Connection-Mode networking services (CMNS) information
  .
  .
  .
  x25                X.25 information
```

Using the no and default Forms of Commands

Almost every configuration command has a **no** form. In general, use the **no** form to disable a feature or function. Use the command without the **no** keyword to reenable a disabled feature or to enable a feature that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** form of the **ip routing** command. To reenable it, use the plain **ip routing** form. The Cisco IOS software command reference publications describe the function of the **no** form of the command whenever a **no** form is available.

Many CLI commands also have a **default** form. By issuing the **default** *command-name command*, you can configure the command to its default setting. The Cisco IOS software command reference documents generally describe the function of the **default** form of the command when the **default** form performs a different function than the plain and **no** forms of the command. To see what default commands are available on your system, enter **default ?** in the appropriate command mode.

Using Command History

The Cisco IOS CLI provides a history or record of commands that you have entered. This feature is particularly useful for recalling long or complex commands or entries, including access lists. To use the command history feature, perform any of the tasks described in the following sections:

- Setting the Command History Buffer Size, page 15
- Recalling Commands, page 16
- Disabling the Command History Feature, page 16

Setting the Command History Buffer Size

By default, the system records ten command lines in its history buffer. To set the number of command lines that the system will record during the current terminal session, use the following command in privileged EXEC mode:

Command	Purpose
`Router#` **terminal history** [**size** *number-of-lines*]	Enables the command history feature for the current terminal session.

The **no terminal history size** command resets the number of lines saved in the history buffer to the default of ten lines.

To configure the number of command lines the system will record for all sessions on a particular line, use the following command in privileged EXEC mode:

Command	Purpose
`Router(config-line)#` **history** [**size** *number-of-lines*]	Enables the command history feature.

Recalling Commands

To recall commands from the history buffer, use one of the following commands or key combinations:

Command or Key Combination	Purpose
Ctrl-P or the **Up Arrow** key.[1]	Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Ctrl-N or the **Down Arrow** key.[1]	Returns to more recent commands in the history buffer after recalling commands with Ctrl-P or the Up Arrow key. Repeat the key sequence to recall successively more recent commands.
`Router>` **`show history`**	While in user EXEC mode, lists the last several commands entered.

1. The arrow keys function only on American National Standards Institute (ANSI)-compatible terminals.

Disabling the Command History Feature

The command history feature is automatically enabled. To disable it during the current terminal session, use the following command in user EXEC mode:

Command	Purpose
`Router>` **`no terminal history`**	Disables command history for the current session.

To configure a specific line so that the command history feature is disabled, use the following command privileged EXEC mode:

Command	Purpose
`Router(config-line)#` **`no history`**	Disables command history for the line.

Using CLI Editing Features and Shortcuts

A variety of shortcuts and editing features are enabled for the Cisco IOS CLI. The following subsections describe these features:

- Moving the Cursor on the Command Line, page 17
- Completing a Partial Command Name, page 17
- Recalling Deleted Entries, page 18
- Editing Command Lines that Wrap, page 19
- Deleting Entries, page 18
- Continuing Output at the --More-- Prompt, page 19
- Redisplaying the Current Command Line, page 19
- Transposing Mistyped Characters, page 20

Moving the Cursor on the Command Line

Table 2 shows the key combinations or sequences you can use to move the cursor on the command line to make corrections or changes. Ctrl indicates the Control key, which must be pressed simultaneously with its associated letter key. Esc indicates the Escape key, which must be pressed first, followed by its associated letter key. Keys are not case sensitive. Many letters used for CLI navigation and editing were chosen to provide an easy way of remembering their functions. In Table 2 characters are bolded in the "Function Summary" column to indicate the relation between the letter used and the function.

Table 2 Key Combinations Used to Move the Cursor

Keystrokes	Function Summary	Function Details
Left Arrow or **Ctrl-B**	**B**ack character	Moves the cursor one character to the left. When you enter a command that extends beyond a single line, you can press the Left Arrow or Ctrl-B keys repeatedly to scroll back toward the system prompt and verify the beginning of the command entry, or you can press the Ctrl-A key combination.
Right Arrow or **Ctrl-F**	**F**orward character	Moves the cursor one character to the right.
Esc, **B**	**B**ack word	Moves the cursor back one word.
Esc, **F**	**F**orward word	Moves the cursor forward one word.
Ctrl-A	Beginning of line	Moves the cursor to the beginning of the line.
Ctrl-E	**E**nd of line	Moves the cursor to the end of the command line.

Completing a Partial Command Name

If you cannot remember a complete command name, or if you want to reduce the amount of typing you have to perform, enter the first few letters of the command, then press the Tab key. The command line parser will complete the command if the string entered is unique to the command mode. If your keyboard does not have a Tab key, press **Ctrl-I** instead.

The CLI will recognize a command once you have entered enough characters to make the command unique. For example, if you enter **conf** in privileged EXEC mode, the CLI will be able to associate your entry with the **configure** command, because only the **configure** command begins with **conf**.

In the following example the CLI recognizes the unique string for privileged EXEC mode of **conf** when the Tab key is pressed:

```
Router# conf<Tab>
Router# configure
```

When you use the command completion feature the CLI displays the full command name. The command is not executed until you use the Return or Enter key. This way you can modify the command if the full command was not what you intended by the abbreviation. If you enter a set of characters that could indicate more than one command, the system beeps to indicate that the text string is not unique.

If the CLI cannot complete the command, enter a question mark (**?**) to obtain a list of commands that begin with that set of characters. Do not leave a space between the last letter you enter and the question mark (**?**).

For example, entering **co?** will list all commands available in the current command mode:

```
Router# co?
configure connect copy
Router# co
```

Note that the characters you enter before the question mark appear on the screen to allow you to complete the command entry.

Deleting Entries

Use any of the following keys or key combinations to delete command entries if you make a mistake or change your mind:

Keystrokes	Purpose
Delete or **Backspace**	Deletes the character to the left of the cursor.
Ctrl-D	Deletes the character at the cursor.
Ctrl-K	Deletes all characters from the cursor to the end of the command line.
Ctrl-U or **Ctrl-X**	Deletes all characters from the cursor to the beginning of the command line.
Ctrl-W	Deletes the word to the left of the cursor.
Esc, **D**	Deletes from the cursor to the end of the word.

Recalling Deleted Entries

The CLI stores commands or keywords that you delete in a history buffer. Only character strings that begin or end with a space are stored in the buffer; individual characters that you delete (using Backspace or Ctrl-D) are not stored. The buffer stores the last ten items that have been deleted using Ctrl-K, Ctrl-U, or Ctrl-X. To recall these items and paste them in the command line, use the following key combinations:

Keystrokes	Purpose
Ctrl-Y	Recalls the most recent entry in the buffer (press keys simultaneously).
Esc, **Y**	Recalls the previous entry in the history buffer (press keys sequentially).

Note that the Esc, Y key sequence will not function unless you press the Ctrl-Y key combination first. If you press Esc, Y more than ten times, you will cycle back to the most recent entry in the buffer.

Editing Command Lines that Wrap

The CLI provides a wrap-around feature for commands that extend beyond a single line on the screen. When the cursor reaches the right margin, the command line shifts ten spaces to the left. You cannot see the first ten characters of the line, but you can scroll back and check the syntax at the beginning of the command. To scroll back, press Ctrl-B or the Left Arrow key repeatedly until you scroll back to the beginning of the command entry, or press Ctrl-A to return directly to the beginning of the line.

In the following example, the **access-list** command entry extends beyond one line. When the cursor first reaches the end of the line, the line is shifted ten spaces to the left and redisplayed. The dollar sign ($) indicates that the line has been scrolled to the left. Each time the cursor reaches the end of the line, the line is again shifted ten spaces to the left.

```
Router(config)# access-list 101 permit tcp 172.31.134.5 255.255.255.0 172.31.1
Router(config)# $ 101 permit tcp 172.31.134.5 255.255.255.0 172.31.135.0 255.25
Router(config)# $t tcp 172.31.134.5 255.255.255.0 172.31.135.0 255.255.255.0 eq
Router(config)# $31.134.5 255.255.255.0 172.31.135.0 255.255.255.0 eq 45
```

When you have completed the entry, press **Ctrl-A** to check the complete syntax before pressing the Return key to execute the command. The dollar sign ($) appears at the end of the line to indicate that the line has been scrolled to the right:

```
Router(config)# access-list 101 permit tcp 172.31.134.5 255.255.255.0 172.31.1$
```

The Cisco IOS software assumes you have a terminal screen that is 80 columns wide. If you have a different screen-width, use the **terminal width** user EXEC command to set the width of your terminal.

Use line wrapping in conjunction with the command history feature to recall and modify previous complex command entries. See the "Recalling Commands" section in this chapter for information about recalling previous command entries.

Continuing Output at the --More-- Prompt

When you use the Cisco IOS CLI, output often extends beyond the visible screen length. For cases where output continues beyond the bottom of the screen, such as with the output of many **?**, **show**, or **more** commands, the output is paused and a --More-- prompt appears at the bottom of the screen. To resume output, press the Return key to scroll down one line, or press the Spacebar to display the next full screen of output.

Tip If output is pausing on your screen, but you do not see the --More-- prompt, try entering a lower value for the screen length using the **length** line configuration command or the **terminal length** privileged EXEC mode command. Command output will not be paused if the **length** value is set to zero.

For information about filtering output from the --More-- prompt, see the "Searching and Filtering CLI Output" section in this chapter.

Redisplaying the Current Command Line

If you are entering a command and the system suddenly sends a message to your screen, you can easily recall your current command line entry. To redisplay the current command line (refresh the screen), use either of the following key combinations:

Keystrokes	Purpose
Ctrl-L or **Ctrl-R**	Redisplays the current command line.

Transposing Mistyped Characters

If you have mistyped a command entry, you can transpose the mistyped characters. To transpose characters, use the following key combination:

Keystrokes	Purpose
Ctrl-T	Transposes the character to the left of the cursor with the character located to the right of the cursor.

Controlling Capitalization

You can capitalize or lowercase words or capitalize a set of letters with simple key sequences. Note, however, that Cisco IOS commands are generally case-insensitive, and are typically all in lowercase. To change the capitalization of commands, use any of the following key sequences:

Keystrokes	Purpose
Esc, **C**	Capitalizes the letter at the cursor.
Esc, **L**	Changes the word at the cursor to lowercase.
Esc, **U**	Capitalizes letters from the cursor to the end of the word.

Designating a Keystroke as a Command Entry

You can configure the system to recognize a particular keystroke (key combination or sequence) as command aliases. In other words, you can set a keystroke as a shortcut for executing a command. To enable the system to interpret a keystroke as a command, use the either of the following key combinations before entering the command sequence:

Keystrokes	Purpose
Ctrl-V or **Esc**, **Q**	Configures the system to accept the following keystroke as a user-configured command entry (rather than as an editing command).

Disabling and Reenabling Editing Features

The editing features described in the previous sections were introduced in Cisco IOS Release 9.21, and are automatically enabled on your system. However, there may be some unique situations that could warrant disabling these editing features. For example, you may have scripts that conflict with editing functionality. To globally disable editing features, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# no editing`	Disables CLI editing features for a particular line.

To disable the editing features for the current terminal session, use the following command in user EXEC mode:

Command	Purpose
`Router# no terminal editing`	Disables CLI editing features for the local line.

To reenable the editing features for the current terminal session, use the following command in user EXEC mode:

Command	Purpose
`Router# terminal editing`	Enables the CLI editing features for the current terminal session.

To reenable the editing features for a specific line, use the following command user EXEC mode:

Command	Purpose
`Router(config-line)# editing`	Enables the CLI editing features.

Searching and Filtering CLI Output

The Cisco IOS CLI provides ways of searching through large amounts of command output and filtering output to exclude information you do not need. These features are enabled for **show** and **more** commands, which generally display large amounts of data.

Note **Show** and **more** commands are always entered in user EXEC or privileged EXEC.

When output continues beyond what is displayed on your screen, the Cisco IOS CLI displays a --More-- prompt. Pressing Return displays the next line; pressing the Spacebar displays the next screen of output. The CLI String Search feature allows you to search or filter output from --More-- prompts.

Understanding Regular Expressions

A regular expression is a pattern (a phrase, number, or more complex pattern) the CLI String Search feature matches against **show** or **more** command output. Regular expressions are case-sensitive and allow for complex matching requirements. Simple regular expressions include entries like `Serial`, `misses`, or `138`. Complex regular expressions include entries like `00210...`, `( is )`, or `[Oo]utput`.

A regular expression can be a single-character pattern or a multiple-character pattern. That is, a regular expression can be a single character that matches the same single character in the command output or multiple characters that match the same multiple characters in the command output. The pattern in the command output is referred to as a string. This section describes creating both single-character patterns and multiple-character patterns. It also discusses creating more complex regular expressions using multipliers, alternation, anchoring, and parentheses.

Single-Character Patterns

The simplest regular expression is a single character that matches the same single character in the command output. You can use any letter (A–Z, a–z) or digit (0–9) as a single-character pattern. You can also use other keyboard characters (such as ! or ~) as single-character patterns, but certain keyboard characters have special meaning when used in regular expressions. Table 3 lists the keyboard characters that have special meaning.

Table 3 Characters with Special Meaning

Character	Special Meaning
.	Matches any single character, including white space.
*	Matchers 0 or more sequences of the pattern.
+	Matches 1 or more sequences of the pattern.
?	Matches 0 or 1 occurrences of the pattern.
^	Matches the beginning of the string.
$	Matches the end of the string.
_ (underscore)	Matches a comma (,), left brace ({), right brace (}), left parenthesis (( ), right parenthesis ( )), the beginning of the string, the end of the string, or a space.

To use these special characters as single-character patterns, remove the special meaning by preceding each character with a backslash (\). The following examples are single-character patterns matching a dollar sign, an underscore, and a plus sign, respectively.

\$ _ \+

You can specify a range of single-character patterns to match against command output. For example, you can create a regular expression that matches a string containing one of the following letters: a, e, i, o, or u. Only one of these characters must exist in the string for pattern matching to succeed. To specify a range of single-character patterns, enclose the single-character patterns in square brackets ([ ]). For example, **[aeiou]** matches any one of the five vowels of the lowercase alphabet, while **[abcdABCD]** matches any one of the first four letters of the lower- or uppercase alphabet.

You can simplify ranges by entering only the endpoints of the range separated by a dash (-). Simplify the previous range as follows:

[a-dA-D]

To add a dash as a single-character pattern in your range, include another dash and precede it with a backslash:

[a-dA-D\-]

You can also include a right square bracket (]) as a single-character pattern in your range, as shown here:

[a-dA-D\-\]]

The previous example matches any one of the first four letters of the lower- or uppercase alphabet, a dash, or a right square bracket.

You can reverse the matching of the range by including a caret (^) at the start of the range. The following example matches any letter except the ones listed:

[^a-dqsv]

The following example matches anything except a right square bracket (]) or the letter d:

[^\]d]

Multiple-Character Patterns

When creating regular expressions, you can also specify a pattern containing multiple characters. You create multiple-character regular expressions by joining letters, digits, or keyboard characters that do not have special meaning. For example, a4% is a multiple-character regular expression. Insert a backslash before the keyboard characters that have special meaning when you want to indicate that the character should be interpreted literally.

With multiple-character patterns, order is important. The regular expression **a4%** matches the character a followed by a 4 followed by a % sign. If the string does not have a4%, in that order, pattern matching fails. The multiple-character regular expression **a.** uses the special meaning of the period character to match the letter a followed by any single character. With this example, the strings ab, a!, or a2 are all valid matches for the regular expression.

You can remove the special meaning of the period character by inserting a backslash before it. For example, when the expression **a\.** is used in the command syntax, only the string a. will be matched.

You can create a multiple-character regular expression containing all letters, all digits, all keyboard characters, or a combination of letters, digits, and other keyboard characters. For example, **telebit 3107 v32bis** is a valid regular expression.

Multipliers

You can create more complex regular expressions that instruct Cisco IOS software to match multiple occurrences of a specified regular expression. To do so, you use some special characters with your single-character and multiple-character patterns. Table 4 lists the special characters that specify "multiples" of a regular expression.

Table 4 Special Characters Used as Multipliers

Character	Description
*	Matches 0 or more single-character or multiple-character patterns.
+	Matches 1 or more single-character or multiple-character patterns.
?	Matches 0 or 1 occurrences of a single-character or multiple-character pattern.

The following example matches any number of occurrences of the letter a, including none:

a*

The following pattern requires that at least one letter a be in the string to be matched:

a+

The following pattern matches the string bb or bab:

ba?b

The following string matches any number of asterisks (*):

*****

To use multipliers with multiple-character patterns, you enclose the pattern in parentheses. In the following example, the pattern matches any number of the multiple-character string ab:

(ab)*

As a more complex example, the following pattern matches one or more instances of alphanumeric pairs, but not none (that is, an empty string is not a match):

([A-Za-z][0-9])+

The order for matches using multipliers (*, +, or ?) is to put the longest construct first. Nested constructs are matched from outside to inside. Concatenated constructs are matched beginning at the left side of the construct. Thus, the regular expression matches A9b3, but not 9Ab3 because the letters are specified before the numbers.

Alternation

Alternation allows you to specify alternative patterns to match against a string. You separate the alternative patterns with a vertical bar (|). Exactly one of the alternatives can match the string. For example, the regular expression **codex|telebit** matches the string codex or the string telebit, but not both codex and telebit.

Anchoring

You can instruct Cisco IOS software to match a regular expression pattern against the beginning or the end of the string. That is, you can specify that the beginning or end of a string contain a specific pattern. You "anchor" these regular expressions to a portion of the string using the special characters shown in Table 5.

Table 5 Special Characters Used for Anchoring

Character	Description
^	Matches the beginning of the string.
$	Matches the end of the string.

For example, the regular expression **^con** matches any string that starts with con, and **$sole** matches any string that ends with sole.

In addition to indicating the beginning of a string, the ^ symbol can be used to indicate the logical function "not" when used in a bracketed range. For example, the expression **[^abcd]** indicates a range that matches any single letter, as long as it is not the letters a, b, c, or d.

Contrast these anchoring characters with the special character underscore (_). Underscore matches the beginning of a string (^), the end of a string ($), parentheses (( )), space ( ), braces ({ }), comma (,), or underscore (_). With the underscore character, you can specify that a pattern exist anywhere in the string. For example, **_1300_** matches any string that has 1300 somewhere in the string. The string 1300 can be preceded by or end with a space, brace, comma, or underscore. So, although {1300_ matches the regular expression **_1300_**, 21300 and 13000 do not.

Using the underscore character, you can replace long regular expression lists. For example, instead of specifying **^1300( ) ( )1300$ {1300, ,1300, {1300} ,1300, (1300** you can specify simply **_1300_**.

Parentheses for Recall

As shown in the “Multipliers” section, you use parentheses with multiple-character regular expressions to multiply the occurrence of a pattern. You can also use parentheses around a single- or multiple-character pattern to instruct the Cisco IOS software to remember a pattern for use elsewhere in the regular expression.

To create a regular expression that recalls a previous pattern, you use parentheses to indicate memory of a specific pattern and a backslash (\) followed by a number to reuse the remembered pattern. The number specifies the occurrence of a parentheses in the regular expression pattern. If you have more than one remembered pattern in your regular expression, then \1 indicates the first remembered pattern, and \2 indicates the second remembered pattern, and so on.

The following regular expression uses parentheses for recall:

a(.)bc(.)\1\2

This regular expression matches an a followed by any character (call it character no. 1), followed by bc followed by any character (character number 2), followed by character no. 1 again, followed by character number. 2 again. So, the regular expression can match aZbcTZT. The software remembers that character number 1 is Z and character number 2 is T and then uses Z and T again later in the regular expression.

Searching and Filtering show Commands

To search **show** command output, use the following command in privileged EXEC mode:

Command	Purpose
`Router#` **`show`** *`any-command`* **`|`** **`begin`** *`regular-expression`*	Begins unfiltered output of the **show** command with the first line that contains the regular expression.

Note Cisco IOS documentation generally uses the vertical bar to indicate a choice of syntax. However, to search the output of **show** and **more** commands, you will need to enter the pipe character (the vertical bar). In this section the pipe appears in bold (**|**) to indicate that you should enter this character.

To filter **show** command output, use one of the following commands in privileged EXEC mode:

Command	Purpose
`Router#` **`show`** *`any-command`* **`|`** **`exclude`** *`regular-expression`*	Displays output lines that do not contain the regular expression.
`Router#` **`show`** *`any-command`* **`|`** **`include`** *`regular-expression`*	Displays output lines that contain the regular expression.

On most systems you can enter the Ctrl-Z key combination at any time to interrupt the output and return to privileged EXEC mode. For example, you can enter the **show running-config | begin hostname** command to start the display of the running configuration file at the line containing the hostname setting, then use Ctrl-Z when you get to the end of the information you are interested in.

Searching and Filtering more Commands

You can search **more** commands the same way you search **show** commands (**more** commands perform the same function as **show** commands). To search **more** command output, use the following command in user EXEC mode:

Command	Purpose
`Router#` **`more`** *`any-command`* **`|`** **`begin`** *`regular-expression`*	Begins unfiltered output of a **more** command with the first line that contains the regular expression.

You can filter **more** commands the same way you filter **show** commands. To filter **more** command output, use one of the following commands in user EXEC mode:

Command	Purpose
`Router#` **`more`** *`any-command`* **`|`** **`exclude`** *`regular-expression`*	Displays output lines that do not contain the regular expression.
`Router#` **`more`** *`any-command`* **`|`** **`include`** *`regular-expression`*	Displays output lines that contain the regular expression.

Searching and Filtering from the --More-- Prompt

You can search output from --More-- prompts. To search **show** or **more** command output from a --More-- prompt, use the following command in user EXEC mode:

Command	Purpose
`-More-` **`/`***`regular-expression`*	Begins unfiltered output with the first line that contains the regular expression.

You can filter output from --More-- prompts. However, you can specify only one filter for each command. The filter remains until the **show** or **more** command output finishes or until you interrupt the output (using Ctrl-Z or Ctrl-6). Therefore, you cannot add a second filter at a --More-- prompt if you already specified a filter at the original command or at a previous --More--prompt.

Note Searching and filtering are different functions. You can search command output using the **begin** keyword and specify a filter at the --More-- prompt for the same command.

To filter **show** or **more** command output at a --More-- prompt, use one of the following commands in user EXEC mode:

Command	Purpose
`-More-` **`-`***`regular-expression`*	Displays output lines that do not contain the regular expression.
`-More-` **`+`***`regular-expression`*	Displays output lines that contain the regular expression.

Using the Cisco IOS CLI: Examples

The following sections provide examples of using the CLI:

- Determining Command Syntax and Using Command History: Example, page 27
- Searching and Filtering CLI Output: Examples, page 28

Determining Command Syntax and Using Command History: Example

The CLI provides error isolation in the form of an error indicator, a caret symbol (^). The ^ symbol appears at the point in the command string where you have entered an incorrect command, keyword, or argument.

In the following example, suppose you want to set the clock. Use context-sensitive help to determine the correct command syntax for setting the clock.

```
Router# clock ?
  set  Set the time and date
Router# clock
```

The help output shows that the **set** keyword is required. Determine the syntax for entering the time:

```
Router# clock set ?
hh:mm:ss   Current time
Router# clock set
```

Enter the current time:

```
Router# clock set 13:32:00
% Incomplete command.
```

The system indicates that you need to provide additional arguments to complete the command. Press Ctrl-P or the Up Arrow to automatically repeat the previous command entry. Then add a space and question mark (**?**) to reveal the additional arguments:

```
Router# clock set 13:32:00 ?
  <1-31>     Day of the month
  January    Month of the year
  February
  March
  April
  May
  June
  July
  August
  September
  October
  November
  December
```

Now you can complete the command entry:

```
Router# clock set 13:32:00 23 February 01
                                       ^
% Invalid input detected at '^' marker.
```

The caret symbol (^) and help response indicate an error at 01. To list the correct syntax, enter the command up to the point where the error occurred and then enter a question mark (**?**):

```
Router# clock set 13:32:00 23 February ?
  <1993-2035> Year
```

```
Router# clock set 13:32:00 23 February
```

Enter the year using the correct syntax and press Enter or Return to execute the command:

Router# **clock set 13:32:00 23 February 2001**

Searching and Filtering CLI Output: Examples

The following is partial sample output from the **more nvram:startup-config | begin** privileged Exec mode command that begins unfiltered output with the first line that contains the regular expression ip. At the --More-- prompt, the user specifies a filter to exclude output lines that contain the regular expression ip.

```
Router# more nvram:startup-config | begin ip
ip subnet-zero
ip domain-name cisco.com
ip name-server 192.168.48.48
ip name-server 172.16.2.132
!
isdn switch-type primary-5ess
.
.
.
interface Ethernet1
 ip address 10.5.5.99 10.255.255.0
 --More--
-ip
filtering...
 media-type 10BaseT
!
interface Serial0:23
 encapsulation frame-relay
 no keepalive
 dialer string 4001
 dialer-group 1
 isdn switch-type primary-5ess
 no fair-queue
```

The following is partial sample output of the **more nvram:startup-config | include** privileged EXEC command. It only displays lines that contain the regular expression `ip`.

```
Router# more nvram:startup-config | include ip
ip subnet-zero
ip domain-name cisco.com
ip name-server 1192.168.48.48
ip name-server 172.16.2.132
```

The following is partial sample output from the **more nvram:startup-config | exclude** privileged EXEC command. It excludes lines that contain the regular expression `service`. At the --More-- prompt, the user specifies a filter with the regular expression Dialer1. Specifying this filter resumes the output with the first line that contains Dialer1.

```
Router# more nvram:startup-config | exclude service
!
version 12.2
!
hostname router
!
boot system flash
no logging buffered
```

```
!
ip subnet-zero
ip domain-name cisco.com
.
.
.
--More--
/Dialer1
filtering...
interface Dialer1
 no ip address
 no ip directed-broadcast
 dialer in-band
 no cdp enable
```

The following is partial sample output from the **show interface** user EXEC or privileged EXEC command mode with an output search specified. The use of the keywords **begin Ethernet** after the pipe begins unfiltered output with the first line that contains the regular expression `Ethernet`. At the --More-- prompt, the user specifies a filter that displays only the lines that contain the regular expression `Serial`.

```
Router# show interface | begin Ethernet

Ethernet0 is up, line protocol is up
Hardware is Lance, address is 0060.837c.6399 (bia 0060.837c.6399)
  Description: ip address is 172.1.2.14 255.255.255.0
  Internet address is 172.1.2.14/24
.
.
.
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
--More--
+Serial
filtering...
Serial1 is up, line protocol is up
Serial2 is up, line protocol is up
Serial3 is up, line protocol is down
Serial4 is down, line protocol is down
Serial5 is up, line protocol is up
Serial6 is up, line protocol is up
Serial7 is up, line protocol is up
```

The following is partial sample output from the **show buffers | exclude** command. It excludes lines that contain the regular expression `0 misses`. At the --More-- prompt, the user specifies a search that continues the filtered output beginning with the first line that contains Serial0.

```
Router# show buffers | exclude 0 misses

Buffer elements:
     398 in free list (500 max allowed)
Public buffer pools:
Small buffers, 104 bytes (total 50, permanent 50):
     50 in free list (20 min, 150 max allowed)
     551 hits, 3 misses, 0 trims, 0 created
Big buffers, 1524 bytes (total 50, permanent 50):
     49 in free list (5 min, 150 max allowed)
Very Big buffers, 4520 bytes (total 10, permanent 10):
.
.
.
Huge buffers, 18024 bytes (total 0 permanent 0):
     0 in free list (0 min, 4 max allowed)
```

```
--More--
/Serial0
filtering...
Serial0 buffers, 1543 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
```

The following is partial sample output from the **show interface | include** user EXEC or privileged EXEC command mode. The use of the **include (is)** keywords after the pipe (|) causes the command to display only lines that contain the regular expression (is). The parenthesis force the inclusion of the spaces before and after is. Use of the parenthesis ensures that only lines containing is with a space both before and after it will be included in the output (excluding from the search, for example, words like "disconnect").

```
router# show interface | include ( is )

ATM0 is administratively down, line protocol is down
  Hardware is ATMizer BX-50
Dialer1 is up (spoofing), line protocol is up (spoofing)
  Hardware is Unknown
  DTR is pulsed for 1 seconds on reset
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 0060.837c.6399 (bia 0060.837c.6399)
  Internet address is 172.21.53.199/24
Ethernet1 is up, line protocol is up
  Hardware is Lance, address is 0060.837c.639c (bia 0060.837c.639c)
  Internet address is 10.5.5.99/24
Serial0:0 is down, line protocol is down
  Hardware is DSX1
.
.
.
 --More--
```

At the --More-- prompt, the user specifies a search that continues the filtered output beginning with the first line that contains Serial0:13:

```
/Serial0:13
filtering...
Serial0:13 is down, line protocol is down
  Hardware is DSX1
  Internet address is 10.0.0.2/8
     0 output errors, 0 collisions, 2 interface resets
  Timeslot(s) Used:14, Transmitter delay is 0 flag
```

EXEC Commands in Configuration Mode

Feature History

Release	Modification
12.1(11b)E, 12.2(7)B, 12.2(7)PB, 12.0(20)SP, 12.0(20)ST, 12.0(21)S, 12.2(8)T	This feature (the **do** command) was introduced.

This document describes the EXEC Commands in Configuration Mode feature and contains the following sections:

Feature Overview

You can now issue EXEC-level Cisco IOS commands (such as **show**, **clear**, and **debug** commands) from within any configuration mode (such as global configuration mode) by issuing the **do** command followed by the desired EXEC command.

Benefits

This feature provides the convenience of entering EXEC-level commands without needing to exit the current configuration mode.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions

You cannot use the **do** command to execute the **configure terminal** EXEC command because issuing the **configure terminal** command changes the mode to configuration mode.

Related Documents

- *Cisco IOS Configuration Fundamentals Configuration Guide*
- *Cisco IOS Configuration Fundamentals Command Reference*

Supported Platforms

- This command is supported on all platforms running the software releases (and all derivative releases) listed in the Feature History at the beginning of this document.

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

No new or modified RFCs are supported by this feature.

Configuration Tasks

See the following section for the configuration task for the EXEC Commands in Configuration Mode feature:

- Executing an EXEC Command in Configuration Mode (optional)

Executing an EXEC Command in Configuration Mode

To execute an EXEC-level command in any configuration mode (including configuration submodes), issue the following command in global configuration mode or the mode from which you want to issue the EXEC command:

Command	Purpose
`Router(config)# do command` `Router(config)#` or `Router(config-if)# do command` `Router(config-if)#`	Allows you to execute any EXEC mode command from within any configuration mode. • *command*—The EXEC command to be executed.

Configuration Examples

This section provides the following configuration examples:

- Executing an EXEC Command in Configuration Mode Examples

Executing an EXEC Command in Configuration Mode Examples

The following example shows how to execute the EXEC-level **show interface** command from within global configuration mode:

```
Router(config)# do show interfaces serial 3/0

Serial3/0 is up, line protocol is up
  Hardware is M8T-RS232
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input never, output 1d17h, output hang never
  Last clearing of "show interface" counters never
.
.
.
Router(config)#
```

The following example shows how to execute the EXEC-level **clear vpdn tunnel** command from within VPDN configuration mode:

```
Router(config-vpdn)# do clear vpdn tunnel
Router(config-vpdn)#
```

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **do**

Show Command Output Redirection

Last Updated: May 2, 2008

This feature adds the capability to redirect output from Cisco IOS command-line interface (CLI) **show** commands and **more** commands to a file.

Feature Specifications for the Show Command Output Redirection Feature

Feature History	
Release	**Modification**
12.0(21)S	This feature was introduced.
12.2(13)T	This feature was integrated into Cisco IOS Release 12.2 T.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Contents

Information About Show Command Output Redirection

This feature enhances the **show** commands in the Cisco IOS CLI to allow large amounts of data output to be written directly to a file for later reference. This file can be saved on local or remote storage devices such as Flash, a SAN Disk, or an external memory device.

For each **show** command issued, a new file can be created, or the output can be appended to an existing file. Command output can optionally be displayed on-screen while being redirected to a file by using the **tee** keyword. Redirection is available using a pipe (|) character after any **show** command, combined with the **redirect**, **append**, or **tee** keywords.

These extensions can also be added to **more** commands.

How to Use the Show Command Enhancement

No configuration tasks are associated with this enhancement. For usage guidelines, see the command pages in the "Command Reference" section on page 3.

Additional References

For information about specific **show** and **more** commands, see the Cisco IOS Documentation Set for Release 12.2 T, available on Cisco.com.

No standards, MIBs, or RFCs are applicable to this feature.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and lots more. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Command List, All Releases*.

- **more <url> append**
- **show <command> append**
- **show <command> redirect**
- **show <command> tee**

Configuration Using Setup and Autoinstall

Overview: Basic Configuration of a Cisco Networking Device

First published: August 9, 2005
Last updated: May 2, 2008

Cisco IOS software provides two features, AutoInstall and Setup mode, to simplify configuring a Cisco IOS-based networking device. AutoInstall enables automatic loading of device configuration files from a remote location and can be used to configure several devices concurrently. Setup is an interactive Cisco IOS software command-line interface (CLI) mode that guides you through a basic (also called a startup) configuration but limits you to configuring a single device at a time. AutoInstall is an automatic process for the device that is being configured; Setup is a manual process for the device that is being configured.

This module provides an introduction to each feature and directs you to modules that describe the features in detail and explain how to use them.

The terms *initial configuration* and *startup configuration* are used interchangeably.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Basic Configuration of a Cisco Networking Device

Prerequisites for Cisco IOS AutoInstall

- Using AutoInstall to Remotely Configure Cisco Networking Devices is written specifically for networking devices running Cisco IOS Release 12.4(1) or newer. However most of the information in this document can be used to configure networking devices that support AutoInstall and are not running Cisco IOS release 12.4(1) or newer. The two key differences that you must allow for are:
  - Some Cisco networking devices use BOOTP instead of DHCP to request IP address addresses over LAN interfaces. Enabling BOOTP support on your DHCP server will resolve this issue.
  - Some Cisco networking devices use a DHCP client identifier format that is different from the format used by networking devices running Cisco IOS release 12.4(1) or newer. This document only explains the DHCP client identifier format used by networking devices running Cisco IOS release 12.4(1) or newer. Use the process described in the "Determining the Value for the DHCP Client Identifier Automatically" section in Using AutoInstall to Remotely Configure Cisco Networking Devices to determine the DHCP client identifier format that your Cisco networking device is using.
- No configuration file resides in NVRAM on the networking device that is being configured with AutoInstall.
- The configuration files that you want to load on to the networking device using AutoInstall reside on a TFTP server that is connected to the network. In most cases there is more than one file; for example, a network file with the IP-to-hostname mappings and a device-specific configuration file.
- You have someone at the remote site to connect the networking device that is being configured with AutoInstall to the network and power it on.
- The network has the IP connectivity necessary to permit the networking device to load configuration files from the TFTP server during the AutoInstall process.
- A DHCP server is available on the network to provide IP addresses to networking devices that are using AutoInstall over a LAN connection.

Prerequisites for Cisco IOS Setup Mode

- A terminal is connected to the console port of the device being configured.
- You know the interfaces you want to configure.
- You know the routing protocols you want to enable.

  For information about routing protocols, see the *Cisco IOS IP Routing Protocols Configuration Guide*, Release 12.4.
- You know whether the device you are configuring will perform bridging.
- You know whether the device you are configuring has protocol translation installed.
- You have network addresses for the protocols being configured.

  For information about network addresses, see the *Cisco IOS IP Addressing Services Configuration Guide*, Release 12.4.
- You have a password strategy for your network environment.

  For information about passwords and device security, see "*Configuring Security with Passwords, Privilege Levels, and Login User names for CLI Sessions on Networking Devices*" in the *Cisco IOS Security Configuration Guide*, Release 12.4.

- You have or have access to documentation for the product you want to configure.

Restrictions for Basic Configuration of a Cisco Networking Device

Restrictions for Cisco IOS AutoInstall

- (Serial interfaces only) AutoInstall over a serial interface using either HDLC or Frame Relay can be performed only over the first serial port on a new device (serial interface 0 or serial interface x/0).
- (LAN interfaces only) Only LAN Token Ring interfaces that set ring speed with physical jumpers support AutoInstall.

Restrictions for Cisco IOS Setup Mode

- Setup mode is hardware dependent. You must follow instructions for the specific product you want to configure, as described in documentation for that product.
- Some configuration parameters apply only when a networking device has the protocol translation option. If a device does not have protocol translation, Setup does not prompt for these parameters.

Information About Basic Configuration of a Cisco Networking Device

Before you configure a networking device with a basic configuration, you should understand the following concepts and decide whether AutoInstall or Setup mode is the best method, based on your requirements.

- Comparison of Cisco IOS AutoInstall and Cisco IOS Setup Mode, page 3
- Cisco IOS AutoInstall, page 3
- Cisco IOS Setup Mode, page 4

Comparison of Cisco IOS AutoInstall and Cisco IOS Setup Mode

Cisco IOS AutoInstall enables automatic loading of device configuration files from a remote location and can be used to configure several devices concurrently. Setup is an interactive Cisco IOS software CLI mode that guides you through a basic (also called a startup) configuration but limits you to configuring a single device at a time. AutoInstall is an automatic process; Setup is a manual process.

Cisco IOS AutoInstall

AutoInstall is the Cisco IOS software feature that enables the configuration of a remote networking device from a central location. The configuration files must be stored on a TFTP server that is accessible by the devices that you are using AutoInstall to setup.

AutoInstall is supported over Ethernet, Token Ring, and FDDI interfaces for LANs, serial interfaces using High-Level Data Link Control (HDLC) encapsulation, and serial interfaces using Frame Relay encapsulation for WANs.

AutoInstall is designed to facilitate central management of installations at remote sites. The AutoInstall process begins when a Cisco IOS software-based device is turned on and a valid configuration file is not found in NVRAM. AutoInstall may not start if the networking device has Cisco Router and Security Device Manager (SDM) or Cisco Network Assistant already installed. In this case, to enable AutoInstall you need to disable SDM.

Using AutoInstall to Remotely Configure Cisco Networking Devices describes how AutoInstall functions, how to disable SDM, and how to configure devices to use AutoInstall.

Cisco IOS Setup Mode

Cisco IOS Setup mode enables you to build an initial configuration file using the Cisco IOS CLI or System Configuration Dialog. The dialog guides you through initial configuration and is useful when you are unfamiliar with Cisco products or the CLI and when configuration changes do not require the level of detail the CLI provides.

Setup starts automatically when a device has no configuration file in NVRAM and is not preconfigured from the factory to use Cisco SDM. When setup completes, it presents the System Configuration Dialog. This dialog guides you through an initial configuration with prompts for basic information about your device and network and then creates an initial configuration file. After the file is created, you can use the CLI to perform additional configuration.

Using Setup Mode to Configure a Cisco Networking Device describes how to use Setup to build a basic configuration and to make configuration changes.

Where to Go Next

Proceed to either Using AutoInstall to Remotely Configure Cisco Networking Devices module or Using Setup Mode to Configure a Cisco Networking Device.

Additional References

This section provides references related to the basic configuration of a Cisco networking device.

Related Documents

Related Topic	Document Title
Configuring a networking device for the first time using the Cisco IOS software feature AutoInstall.	Using AutoInstall to Remotely Configure Cisco Networking Devices
Configuring a networking device using Cisco IOS Setup mode	Using Setup Mode to Configure a Cisco Networking Device
Configuration fundamentals and associated commands	*Cisco IOS Configuration Fundamentals Configuration Guide* for your release and the release-independent *Cisco IOS Configuration Fundamentals Command Reference*

Technical Assistance

Description	Link
The Cisco Technical Support and Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Using Setup Mode to Configure a Cisco Networking Device

Setup mode provides an interactive menu to help you to create an initial configuration file for a new networking device, or a device that you have erased the startup-config file from NVRAM. The nteractive menu guides you through initial configuration and is useful when you are unfamiliar with Cisco products or the command line interface (CLI) and when configuration changes do not require the level of detail the CLI provides. Setup mode can also be used to modify an existing configuration.

This module describes how to use the System Configuration Dialog to prepare a Cisco networking device for full configuration and how you can make configuration changes after an initial configuration is complete.

In this module, to improve readability filenames are enclosed in quotation marks. Also, the terms *device* and *networking device* mean a router, switch, or other device running Cisco IOS software. The terms *initial configuration* and *startup configuration* are used interchangeably.

Module History

This module was first published on August 9, 2005, and last updated on October 2006.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Using Cisco IOS Setup Mode to Configure a Cisco Networking Device

- You have read the "Basic Configuration of a Cisco Networking Device Overview" module.
- An ASCII terminal is connected to the console port of the device being configured.
- You know the interfaces you want to configure.
- You know the routing protocols you want to enable.

  For information about routing protocols, see the *Cisco IOS IP Routing Protocols Configuration Guide*, Release 12.4.
- You know whether the device you are configuring will perform bridging.
- You know whether the device you are configuring has protocol translation installed.
- You have network addresses for the protocols being configured.

  For information about network addresses, see the *Cisco IOS IP Addressing Services Configuration Guide*, Release 12.4.
- You have a password strategy for your network environment.

  For information about passwords and device security, see "Configuring Security with Passwords, Privilege Levels, and Login User names for CLI Sessions on Networking Devices" in the *Cisco IOS Security Configuration Guide*, Release 12.4.
- You have or have access to documentation for the product you want to configure.

Restrictions for Using Cisco IOS Setup Mode to Configure a Cisco Networking Device

- Setup mode is hardware dependent. You must follow instructions for the specific product you want to configure, as described in documentation for that product.
- Some configuration parameters apply only when a networking device has the protocol translation option. If a device does not have protocol translation, Setup does not prompt for these parameters.

Information About Using Cisco IOS Setup Mode to Configure a Cisco Networking Device

Before you use Cisco IOS Setup mode to configure a Cisco networking device, you should understand the following concepts:

Cisco IOS Setup Mode

Cisco IOS Setup mode enables you to build an initial configuration file using the Cisco IOS CLI or System Configuration Dialog. The dialog guides you through initial configuration and is useful when you are unfamiliar with Cisco products or the CLI and when configuration changes do not require the level of detail the CLI provides.

Setup starts automatically when a device has no configuration file in NVRAM and is not preconfigured from the factory to use Cisco Router and Security Device Manager (SDM). When setup completes, it presents the System Configuration Dialog. This dialog guides you through an initial configuration with prompts for basic information about your device and network and then creates an initial configuration file. After the file is created, you can use the CLI to perform additional configuration.

Cisco Router and Security Device Manager

Cisco SDM is a web-based device management tool for configuring Cisco IOS network connections and security features on networking devices. SDM provides a default configuration and various wizards to guide you step by step through configuring a Cisco networking device, additional LAN or WAN connections, and VPN connections; creating firewalls; and performing security audits.

In addition to building an initial configuration, SDM provides an Advanced Mode through which you can configure advanced features such as Firewall Policy and Network Address Translation (NAT).

Some Cisco products ship from the factory with SDM installed. If SDM is preinstalled on your device and you want to use Setup to configure an initial configuration, you first must disable the SDM default configuration.

System Configuration Dialog

The *System Configuration Dialog* is an interactive CLI mode that prompts you for information needed to build an initial configuration for a Cisco networking device. Like the CLI, the System Configuration Dialog provides help text at each prompt. To access this help text, you enter a question mark (**?**) at the prompt.

The prompts in the System Configuration Dialog vary depending on hardware, installed interface modules, and software image. To use the dialog for an initial configuration, you need to refer to product-specific documentation.

The values shown in square brackets next to prompts reflect the current settings. These may be default settings from the factory or the latest settings configured on the device. To accept these settings, you press **Enter** on the keyboard.

You can exit (**Ctrl-C**) the System Configuration Dialog and return to privileged EXEC mode without making changes and without going through the entire dialog. If you exit the dialog but want to continue with setup, you can issue the **setup** command in privileged EXEC mode.

When you complete all the steps in the dialog, the device displays the modified configuration file and asks if you want to use that file. You must answer yes or no; there is no default for this prompt. If you answer yes, the file is saved to NVRAM as the startup configuration. If you answer no, the file is not saved and you must start at the beginning of the dialog if you want to build another initial configuration.

In addition to being a quick and easy way to perform an initial configuration, the System Configuration Dialog also is useful for performing basic configuration changes after an initial configuration has been performed.

Benefits of Using Cisco IOS Setup Mode

The System Configuration Dialog in Cisco IOS Setup mode can be a valuable tool for users who are unfamiliar with Cisco products or the CLI. The dialog guides users through the configuration process with prompts for basic information to get the device operational. When general configuration changes are needed, the dialog also is an alternative method to the detail-level CLI.

How to Use Cisco IOS Setup Mode to Configure a Cisco Networking Device and Make Configuration Changes

This section describes how to use the System Configuration Dialog to build an initial configuration file and to make configuration changes after a startup configuration has been loaded.

- Disabling the SDM Default Configuration File, page 4
- Using the System Configuration Dialog to Create an Initial Configuration File, page 5
- Using the System Configuration Dialog to Make Configuration Changes, page 9
- Verifying the Configuration, page 10

Disabling the SDM Default Configuration File

Perform this task if SDM was preinstalled on your device and you want to use Setup to build an initial configuration file. SDM remains on the device.

Perform this task if SDM was pre installed on your device and you want to use AutoInstall to configure the device instead. SDM remains on the device.

SUMMARY STEPS

1. Connect the console cable from the console port on the device to the serial port on the PC.
2. Connect the power supply to the device, plug the power supply into a power outlet, and turn on the device.
3. Connect to the device using a terminal emulation program.
4. **enable**
5. **erase startup-config**
6. **reload**

DETAILED STEPS

Step 1 Connect the console cable, shipped with your device, from the console port on the device to a serial port on your PC. Refer to the hardware installation guide for the device for instructions.

Step 2 Connect the power supply to the device, plug the power supply into a power outlet, and turn on the device. Refer to the quick start guide for the device for instructions.

Step 3 Use Hyperterminal or a similar terminal emulation program on your PC, with the following terminal emulation settings, to connect to the device:

- 9600 baud

- 8 data bits, no parity, 1 stop bit
- No flow control

Step 4 **enable**

Enter privileged EXEC mode.

enable

```
Router> enable
Router#
```

Step 5 **erase startup-config**

Erases the existing configuration in NVRAM.

```
Router# erase startup-config
```

Step 6 **reload**

Initiates the reload process. The router will initiate the AutoInstall process after it finishes the reload process.

```
Router# reload
```

Using the System Configuration Dialog to Create an Initial Configuration File

Perform this task to create an initial configuration for a Cisco networking device.

Prerequisites

If SDM is installed, you must disable its default configuration file before using Setup.

Restrictions

The System Configuration Dialog does not allow you to randomly select or enter parameters for configuration. You must move through the dialog step by step until the screen shows the information you want to change.

SUMMARY STEPS

1. **Power on the device.**
2. **Enter yes at the prompt to enter the initial configuration dialog.**
3. **If you are prompted to continue with the configuration dialogue, enter yes** at **the prompt to continue the dialog (this step might not appear).**
4. Enter **yes** at the prompt to enter basic management setup.
5. **Enter a hostname for the device.**
6. **Enter an enable secret password.**
7. Enter an enable password.
8. Enter a virtual terminal password.

9. Respond to the prompts as appropriate for your network.
10. Select an interface to connect the device to the management console.
11. Respond to the prompts as appropriate for your network.
12. Enter **2** to save the configuration file to NVRAM and exit.

DETAILED STEPS

Step 1 **Power on the device.**

Step 2 **Enter yes at the prompt to enter the initial configuration dialogue.**

If the following messages appear at the end of the startup sequence, the System Configuration Dialog was invoked automatically:

```
--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: yes
```

The screen displays the following:

```
--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]:
```

Step 3 **If you are prompted to continue with the configuration dialogue, enter yes** at **the prompt to continue the dialog (this step might not appear)**.

```
Continue with configuration dialog? [yes/no]: yes
```

Step 4 The basic management screen is displayed:

```
At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system

Would you like to enter basic management setup? [yes/no]:
```

Enter **yes** to enter basic management setup:

```
Would you like to enter basic management setup? [yes/no]: yes

The screen displays the following:

Configuring global parameters:

Enter host name [R1]:
```

Step 5 Enter a hostname for the device. This example uses Router.

```
Configuring global parameters:

Enter host name [R1]: Router

The screen displays the following:
```

```
The enable secret is a password used to protect access to
  privileged EXEC and configuration modes. This password, after
  entered, becomes encrypted in the configuration.
  Enter enable secret:
```

Step 6 Enter an enable secret password. This password is encrypted and cannot be seen when viewing the configuration.

```
Enter enable secret: 1g2j3mm
```

The screen displays the following:

```
The enable password is used when you do not specify an
  enable secret password, with some older software versions, and
  some boot images.
  Enter enable password:
```

Step 7 Enter an enable password that is different from the enable secret password. An enable password is not encrypted and can be seen when viewing the configuration:

```
  Enter enable password: cts54tnl
```

The screen displays the following:

```
The virtual terminal password is used to protect
  access to the router over a network interface.
  Enter virtual terminal password:
```

Step 8 Enter a virtual terminal password. This password allows access to the device through only the console port.

```
  Enter virtual terminal password: tls6gato
```

The screen displays the following:

```
Configure SNMP Network Management? [no]:
```

Step 9 Respond to the following prompts as appropriate for your network. In this example, the current setting [no] is accepted by pressing **Enter**.

```
Configure SNMP Network Management? [no]:
```

A summary of the available interfaces displays. The interface numbering that appears depends on the type of platform and on the installed interface modules and cards.

```
Current interface summary
Interface                  IP-Address      OK? Method Status                Prol
Ethernet0/0                unassigned      YES NVRAM  administratively down dow
Ethernet1/0                unassigned      YES NVRAM  administratively down dow
Serial2/0                  unassigned      YES NVRAM  administratively down dow
Serial3/0                  unassigned      YES NVRAM  administratively down dow
Loopback0                  1.1.1.1         YES NVRAM  up                    up

Enter interface name used to connect to the
management network from the above interface summary:
```

Step 10 Select an interface to connect the router to the management network:

```
Enter interface name used to connect to the
management network from the above interface summary: Ethernet0/0
```

Step 11 Respond to the prompts as appropriate for your network. In this example, IP is configured: an IP address is entered and the current subnet mask is accepted. The screen displays the command script created.

```
Configuring interface Ethernet0/0:
  Configure IP on this interface? [no]: yes
    IP address for this interface: 172.17.1.1
    Subnet mask for this interface [255.255.0.0] :
    Class B network is 172.17.0.0, 16 subnet bits; mask is /16

The following configuration command script was created:

hostname Router
enable secret 5 $1$1Gg9$GuxXfUUBBfVqGvlW4psIm1
enable password cts54tnl
line vty 0 4
password tls6gato
no snmp-server
!
no ip routing
!
interface Ethernet0/0
no shutdown
ip address 172.17.1.1 255.255.0.0
!
interface Ethernet1/0
shutdown
no ip address
!
interface Serial2/0
shutdown
no ip address
!
interface Serial3/0
shutdown
no ip address
!
end

[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.

Enter your selection [2]:
```

Step 12 Enter **2 or press Enter** to save the configuration file to NVRAM and exit.

```
[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.

Enter your selection [2]: 2
```

The screen displays the following:

```
Building configuration...
[OK]
Use the enabled mode 'configure' command to modify this configuration.

Router#
00:01:32: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
00:01:33: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed p
```

What to Do Next

Proceed to the “Verifying the Configuration” section on page 10.

Using the System Configuration Dialog to Make Configuration Changes

The *System Configuration Dialog* is an alternative to the CLI when configuration changes do not require the level of detail the CLI provides. For example, you can use the System Configuration Dialog to add a protocol suite, make addressing scheme changes, or configure a newly installed interface. Although you can use configuration modes available through the CLI to make these changes, the *System Configuration Dialog* provides you a high-level view of the configuration and guides you through the configuration process.

Prerequisites

When you add or modify hardware and need to update a configuration, refer to documentation for your platform for details about physical and logical port assignments.

Restrictions

The System Configuration Dialog does not allow you to randomly select or enter parameters for configuration. You must move through the dialog step by step until the screen shows the information you want to change.

SUMMARY STEPS

1. **enable**
2. **setup**
3. **Follow Steps 3 through 12 in** the **Detailed Steps in the preceding “Using the System Configuration Dialog to Create an Initial Configuration File” section on page 5.**
4. Verify the configuration is modified correctly. Refer to the “Verifying the Configuration” section on page 10.

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode.

```
Router> enable
Router#
```

Step 2 **setup**

The **setup** command puts the router in **setup** mode.

```
Router# setup
```

The screen displays the following:

```
--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]:
```

Enter yes at **the prompt to continue the dialog**.

```
Continue with configuration dialog? [yes/no]: yes

The screen displays the following:

At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system

Would you like to enter basic management setup? [yes/no]:
```

Step 3 **Follow Steps 3 through 12 in** the **Detailed Steps in the preceding "Using the System Configuration Dialog to Create an Initial Configuration File" section on page 5.**

Step 4 Verify the configuration is modified correctly. Refer to the "Verifying the Configuration" section on page 10.

Verifying the Configuration

Perform this task to verify that the configuration you created using the System Configuration Dialog is operating correctly.

SUMMARY STEPS

1. **show interfaces**
2. **show ip interface brief**
3. **show configuration**

DETAILED STEPS

Step 1 **show interfaces**

This command verifies that the interfaces are operating correctly and that they and the line protocol are in the correct state: up or down.

Step 2 **show ip interface brief**

This command displays a summary status of the interfaces configured for IP.

Step 3 **show configuration**

This command verifies that the correct hostname and password were configured.

Examples

This example is the verification of the configuration file created in Steps 1 through 12 of the "Using the System Configuration Dialog to Create an Initial Configuration File" section on page 5.

```
Router# show interfaces
```

```
Ethernet0/0 is up, line protocol is up
  Hardware is AmdP2, address is aabb.cc03.6c00 (bia aabb.cc03.6c00)
  Internet address is 172.17.1.1/16
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:06, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     11 packets output, 1648 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Ethernet1/0 is administratively down, line protocol is down
  Hardware is AmdP2, address is aabb.cc03.6c01 (bia aabb.cc03.6c01)
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Serial2/0 is administratively down, line protocol is down
  Hardware is M4T
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Restart-Delay is 0 secs
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
```

```
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions     DCD=up  DSR=up  DTR=down  RTS=down  CTS=up

Serial3/0 is administratively down, line protocol is down
  Hardware is M4T
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Restart-Delay is 0 secs
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions     DCD=down  DSR=down  DTR=up  RTS=up  CTS=down

Loopback0 is up, line protocol is up
  Hardware is Loopback
  Internet address is 1.1.1.1/32
  MTU 1514 bytes, BW 8000000 Kbit, DLY 5000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation LOOPBACK, loopback not set
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets

Router# show ip interface brief

Interface                  IP-Address      OK? Method Status                Prol
Ethernet0/0                172.17.1.1      YES manual up                    up
Ethernet1/0                unassigned      YES manual administratively down dow
Serial2/0                  unassigned      YES manual administratively down dow
Serial3/0                  unassigned      YES manual administratively down dow
Loopback0                  1.1.1.1         YES NVRAM  up                    up

Router# show configuration
```

```
Using 1029 out of 8192 bytes
!
version 12.3
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$1Gg9$GuxXfUUBBfVqGvlW4psIm1
enable password cts54tnl
!
no aaa new-model
!
resource manager
!
clock timezone PST -8
ip subnet-zero
no ip routing
!
!
!
!
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
 no ip route-cache
!
interface Ethernet0/0
 ip address 172.17.1.1 255.255.0.0
 no ip route-cache
!
interface Ethernet1/0
 no ip address
 no ip route-cache
 shutdown
!
interface Serial2/0
 no ip address
 no ip route-cache
 shutdown
 serial restart-delay 0
!
interface Serial3/0
 no ip address
 no ip route-cache
 shutdown
 serial restart-delay 0
!
!
ip classless
no ip http server
!
!
!
!
control-plane
!
!
line con 0
 transport preferred all
```

```
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password tls6gato
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Configuration Examples for Using Cisco IOS Setup Mode to Configure a Cisco Networking Device

This section provides the following configuration example:

- Configuring Ethernet Interface 0 Using the System Configuration Dialog: Example, page 15

Configuring Ethernet Interface 0 Using the System Configuration Dialog: Example

In the following example, the System Configuration Dialog is used to configure Ethernet interface 0 with an IP address.

Note Prompts and the order in which they appear on the screen vary depending on the platform and the interfaces installed in the device.

```
R1# setup

--- System Configuration Dialog ---
Continue with configuration dialog? [yes/no]: yes
At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system

Would you like to enter basic management setup? [yes/no]: yes
Configuring global parameters:

  Enter host name [R1]: Router

  The enable secret is a password used to protect access to
  privileged EXEC and configuration modes. This password, after
  entered, becomes encrypted in the configuration.
  Enter enable secret: 1g2j3mmc

  The enable password is used when you do not specify an
  enable secret password, with some older software versions, and
  some boot images.
```

```
  Enter enable password: cts54tnl

  The virtual terminal password is used to protect
  access to the router over a network interface.
  Enter virtual terminal password: tls6gato
  Configure SNMP Network Management? [no]:

Current interface summary
Interface                  IP-Address      OK? Method Status                Prol
Ethernet0/0                172.17.1.1      YES manual up                    up
Ethernet1/0                unassigned      YES manual administratively down dow
Serial2/0                  unassigned      YES manual administratively down dow
Serial3/0                  unassigned      YES manual administratively down dow
Loopback0                  1.1.1.1         YES NVRAM  up                    up

Enter interface name used to connect to the
management network from the above interface summary: Ethernet0/0

Configuring interface Ethernet0/0:
  Configure IP on this interface? [no]: yes
    IP address for this interface: 172.17.1.1
    Subnet mask for this interface [255.255.0.0] :
    Class B network is 172.17.0.0, 16 subnet bits; mask is /16

The following configuration command script was created:

hostname Router
enable secret 5 $1$1Gg9$GuxXfUUBBfVqGvlW4psIm1
enable password cts54tnl
line vty 0 4
password tls6gato
no snmp-server
!
no ip routing

!
interface Ethernet0/0
no shutdown
ip address 172.17.1.1 255.255.0.0
!
interface Ethernet1/0
shutdown
no ip address
!
interface Serial2/0
shutdown
no ip address
!
interface Serial3/0
shutdown
no ip address
!
end


[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.

Enter your selection [2]:
Building configuration...
[OK]
Use the enabled mode 'configure' command to modify this configuration.
```

```
Router#
00:01:32: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
00:01:33: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed p
```

Additional References

The following sections provide references related to using Cisco IOS Setup to configure a Cisco networking device.

Related Documents

Related Topic	Document Title
Overview of Cisco IOS Setup Mode and AutoInstall for configuring Cisco networking devices	"Basic Configuration of a Cisco Networking Device Overview"
Configuring a Cisco networking device using the Cisco IOS AutoInstall feature	"Using AutoInstall to Remotely Configure Cisco Networking Devices"

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Using Cisco IOS Setup Mode to Configure a Cisco Networking Device

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or 12.0(3)S or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 1lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Using Cisco IOS Setup Mode to Configure a Cisco Networking Device

Feature Name	Releases	Feature Configuration Information
This table is intentionally left blank because no features were introduced or modified in Cisco IOS Release 12.2(1) or later. This table will be updated when feature information is added to this module.	—	—

Using AutoInstall to Remotely Configure Cisco Networking Devices

First published: November 28, 2005
Last updated: May 2, 2008

AutoInstall enables remote, automatic configuration of networking devices. AutoInstall is typically used to set up new networking devices remotely. You can, however, use AutoInstall to configure existing networking devices after you remove the configuration file from their NVRAM. The AutoInstall process uses pre-existing configuration files that are stored on a TFTP server.

In this module the term *networking device* means a router that runs Cisco IOS software. Also, the following terms are used interchangeably:

- *initial configuration* and *startup configuration*
- *set up* and *configure*

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Using AutoInstall to Remotely Configure a Cisco Networking Device" section on page 52.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Using AutoInstall to Remotely Configure Cisco Networking Devices

- You have read Overview: Basic Configuration of a Cisco Networking Device.
- This document is written specifically for networking devices running Cisco IOS Release 12.4(1) or newer. However most of the information in this document can be used to configure networking devices that support AutoInstall and are not running Cisco IOS release 12.4(1) or newer. The two key differences that you must allow for are:
  - Some Cisco networking devices use BOOTP instead of DHCP to request IP address addresses over LAN interfaces. Enabling BOOTP support on your DHCP server will resolve this issue.
  - Some Cisco networking devices use a DHCP client identifier format that is different from the format used by networking devices running Cisco IOS release 12.4(1) or newer. This document only explains the DHCP client identifier format used by networking devices running Cisco IOS release 12.4(1) or newer. Use the process described in "Determining the Value for the DHCP Client Identifier Automatically" section on page 35 to determine the DHCP client identifier format that your Cisco networking device is using.
- No configuration file resides in NVRAM on the networking device that is being configured with AutoInstall.
- The configuration files that you want to load on to the networking device using AutoInstall reside on a TFTP server that is connected to the network. In most cases there is more than one file; for example, a network file with the IP-to-hostname mappings and a device-specific configuration file.
- You have someone at the remote site to connect the networking device that is being configured with AutoInstall to the network and power it on.
- The network has the IP connectivity necessary to permit the networking device to load configuration files from the TFTP server during the AutoInstall process.
- A DHCP server is available on the network to provide IP addresses to networking devices that are using AutoInstall over a LAN connection.

Restrictions for Using AutoInstall to Remotely Configure Cisco Networking Devices

- (Serial interfaces only) AutoInstall over a serial interface using either HDLC or Frame Relay can be performed only over the first serial port on a new device (serial interface 0 or serial interface x/0).
- (LAN interfaces only) Only LAN Token Ring interfaces that set ring speed with physical jumpers support AutoInstall.

Information About Using AutoInstall to Remotely Configure Cisco Networking Devices

Before you configure or use AutoInstall, you should understand the following information:

- AutoInstall, page 3
- Benefits of Using AutoInstall to Remotely Configure a Cisco Networking Device, page 14

AutoInstall

AutoInstall can be used to load a final full configuration, or a partial temporary configuration, on to a networking device that is being configured with AutoInstall.

Tip When you use AutoInstall to load a partial temporary configuration, you must finish configuring the device manually.

The requirements for provisioning your network for AutoInstall, and the configuration options for provisioning AutoInstall are explained in these sections:

- Services and Servers Used By AutoInstall: Dynamic Assignment of IP Addresses, page 3
- Services and Servers Used By AutoInstall: IP-to-Hostname Mapping, page 7
- Services and Servers Used By AutoInstall: Storage and Transmission of Configuration Files, page 7
- Networking Devices Used by AutoInstall, page 8
- Configuration Files Used by AutoInstall, page 10
- Configuration Options for AutoInstall, page 12
- The AutoInstall Process, page 13

Services and Servers Used By AutoInstall: Dynamic Assignment of IP Addresses

The network must be able to provide the dynamic assignment of an IP address to the networking device that is being configured with AutoInstall. The type of IP address assignment server that is used depends on the type of connection that the networking that is being configured with AutoInstall has to the network.

AutoInstall uses these types of IP address servers:

- DHCP Servers, page 3
- SLARP Servers, page 4
- BOOTP Servers, page 6

DHCP Servers

Networking devices using AutoInstall over a LAN connection require a DHCP server to provide an IP address dynamically. This requirement applies to Ethernet, Token Ring, and FDDI interfaces. The network must be configured to provide IP connectivity between the DHCP server and any devices that are using AutoInstall over LAN connections.

DHCP (defined in RFC 2131) is an extension of the functionality provided by the Bootstrap Protocol (defined in RFC 951). DHCP provides the framework for passing configuration information to hosts on a TCP/IP network. DHCP adds the capability of automatic allocation of reusable network addresses and additional configuration options such as a router (gateway) IP address, a TFTP server IP address, the name of a boot file to load, and the domain name to use. DHCP servers can be configured on routers, UNIX servers, Microsoft Windows-based servers, and other platforms.

DHCP servers typically assign IP addresses from a pool of IP addresses randomly. It is possible for a device that uses DHCP to obtain its IP address to have a different IP address every time it is connected to the network. This creates a problem for the AutoInstall process when you want to ensure that a particular device is assigned a specific hostname during the AutoInstall process. For example, if you are installing routers on different floors in a remote site and each router is supposed to be assigned a name that indicates its location, such as **ChicagoHQ-1st** and **ChicagoHQ-2nd**, you need to ensure that each device gets the IP address that will be mapped to its correct hostname.

The process of ensuring that a device is assigned a specific IP address is referred to as *creating a reservation*. A reservation is a manually configured relationship between an IP address and a physical layer address of a LAN interface on the device. Many Cisco IOS-based devices do not use their MAC address when they request an IP address via DHCP. They use a much longer client identifier instead. Due to the complexity of identifying the client identifier so that you can preconfigure a reservation, and the complexity of finding out if the new device uses its MAC address or the client identifier, we recommend that you allow a new device to obtain an IP address without using a DHCP reservation first in order to discover if the device is using its MAC address or a client identifier. When you have learned how the new device is identifying itself to the DHCP server, you can make a note of the format and create a reservation for it. The next time the new device is rebooted it should obtain the IP address that you reserved to ensure that the new device is assigned the correct hostname. Refer to the information on creating DHCP reservations that was provided with your DHCP server software. The process for creating reservations using Cisco IOS based DHCP servers is explained in the "Using AutoInstall to Set Up Devices Connected to LANs: Example" section on page 31. This section includes instructions for identifying the client identifier before the device is connected to the network so that you can preconfigure the DHCP reservations.

Note This document uses a Cisco router as the DHCP server for using AutoInstall to configure LAN-connected networking devices. If you are using a different device as your DHCP server ensure that you have the user documentation for it available in the event that you need help configuring it.

Note There are several configuration parameters such as TFTP server addresses, DNS server addresses, domain names and so on, that can be provided to LAN-connected clients by DHCP servers during the process of assigning IP addresses to clients. These parameters are not required by AutoInstall, therefore they are not included in this document. If you know how to use these parameters you can include them in your DHCP server configuration when you are using AutoInstall to setup your networking devices.

For more information on DHCP services visit the IETF RFC site (http://www.ietf.org/rfc.html) and look for RFCs about DHCP. Most server operating systems support DHCP servers. Refer to the documentation that was provided with your operating system for more information.

SLARP Servers

A router that is being configured with AutoInstall over a serial interface using HDLC encapsulation will send a Serial Line ARP (SLARP) request for an IP address over the serial interface that is connected to the staging router.

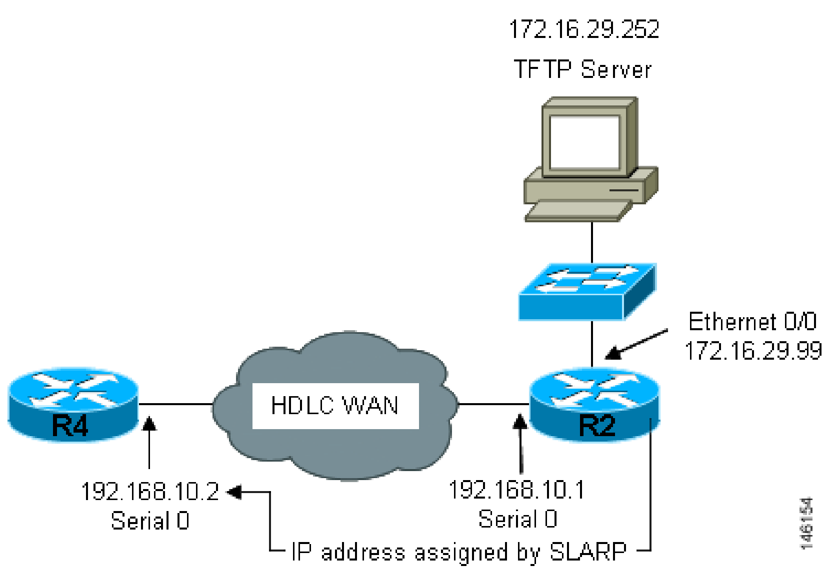
The serial interface of the staging router must be configured with an IP address in which the host portion is 1 or 2, such as 192.168.10.1 or 192.168.10.2. The staging router will send a SLARP response to the router that is being configured with AutoInstall that contains the value that the staging router is not using. For example, if the interface on the staging router that is connected to the router that is being configured with AutoInstall is using 192.168.10.1 as its IP address, the staging router will send a SLARP response with a value of 192.168.10.2 to the router that is being configured with AutoInstall.

Tip If you are using a mask of 255.255.255.252 on the serial interface of the staging router SLARP will assign the available IP host address to the new device. For example, if you assign IP address 198.162.10.5 255.255.255.252 to serial 0 on the staging router, SLARP will assign 198.162.10.6 to the new device. If you assign IP addresses 198.162.10.6 255.255.255.252 to serial 0 on the staging router SLARP will assign 198.162.10.5 to the new device.

Figure 2 shows an example of SLARP.

In Figure 1, the IP address of serial interface 0 on the staging router (R2) is 192.168.10.1. SLARP therefore assigns the IP address 192.168.10.2 to serial interface 0 on the new device.

Figure 1 Using SLARP to Assign an IP Address to a New Device



Note AutoInstall over a serial interface using HDLC can be performed only over the first serial port on a new device (serial interface 0 or serial interface x/0). The staging router and new device must be directly connected using the first serial interface port on the new device; for example, serial 0/0 or if the first serial port is in the second slot of the device, serial 2/0.

Tip The IP address that is assigned to the router that is being configured with AutoInstall by SLARP from the staging router is the IP address that you must use in the **ip host** *hostname ip-address* command in the AutoInstall network-confg or cisconet.cfg file to ensure that the router that is being configured with AutoInstall is assigned the correct hostname so that it can request its host-specific configuration file.

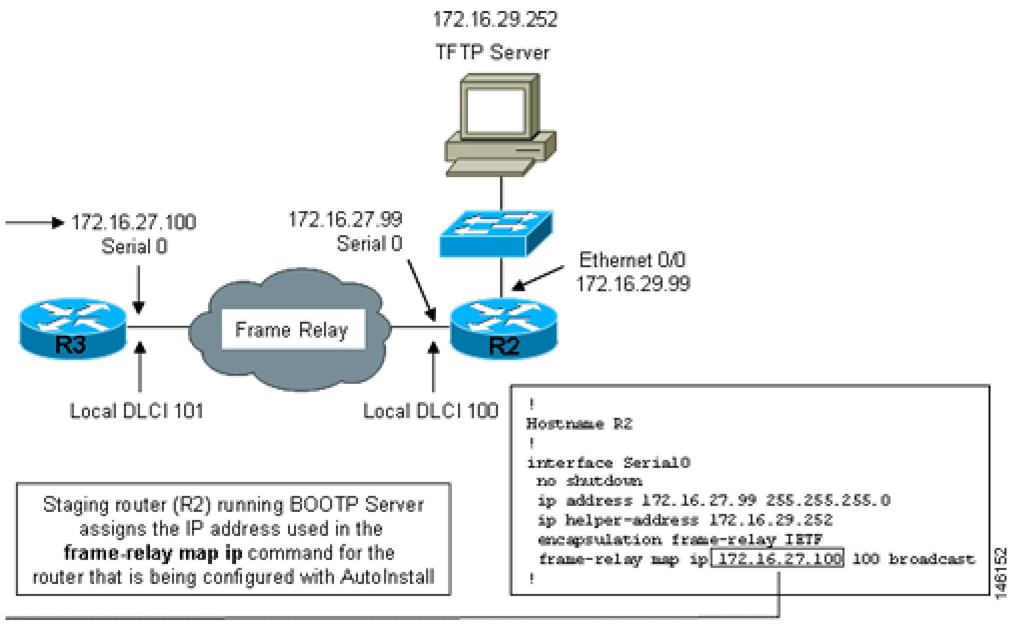
BOOTP Servers

A router that is being configured with AutoInstall over a serial interface using Frame Relay encapsulation will send a BOOTP request for an IP address over the serial interface that is connected to the staging router.

The staging router learns the correct IP address to provide in its BOOTP response to the router that is being configured with AutoInstall by examining the **frame-relay map ip** *ip-address dlci* command that is configured on the interface that it is using to connect to the router that is being configured with AutoInstall.

In Figure 2 R2 is the staging router. R2 has the **frame-relay map ip 172.16.27.100 100** broadcast command configured on interface serial 0. When R2 receives the BOOTP request for an IP address from R3 during the AutoInstall process, R3 will reply with 172.16.27.100.

Figure 2 Example of Using BOOTP for Autoinstall Over a Frame Relay Network



Tip The limitation imposed by SLARP in which the IP addresses for the new device and the staging router must end in either .1 or .2 does not apply to BOOTP. BOOTP for AutoInstall over Frame Relay supports all host addresses for the IP address subnet that is assigned to the Frame Relay circuit between the router that is being configured with AutoInstall and the staging router.

Tip The IP address that is assigned to the router that is being configured with AutoInstall by BOOTP from the staging router is the IP address that you must use in the **ip host** *hostname ip-address* command in the AutoInstall network-confg or cisconet.cfg file to ensure that the router that is being configured with AutoInstall is assigned the correct hostname so that it can request its host-specific configuration file.

Note AutoInstall over a serial interface using Frame Relay encapsulation can be performed only over the first serial port on a new device (serial interface 0 or serial interface x/0). The staging router and new device must be directly connected using the first serial interface port on the new device; for example, serial 0/0 or if the first serial port is in the second slot of the device, serial 2/0.

Services and Servers Used By AutoInstall: IP-to-Hostname Mapping

If you want the networking device to load a full configuration file during the AutoInstall process, the networking device must be able to determine its hostname so that it can request the configuration file that you created specifically for it.

The following caveats apply to the provisioning of IP address to hostname mapping for AutoInstall:

- Any networking device that is being configured with AutoInstall can determine its hostname by loading one of the AutoInstall network configuration files (network-confg or cisconet.cfg) from the TFTP server that contain the **ip host** *hostname ip-address* commands. For example, to map host R3 to IP address 198.162.100.3, the network-confg or cisconet.cfg file must contain the **ip host r3 198.162.100.3** command.
- A networking device that is being configured with AutoInstall over a LAN interface can also determine its hostname by querying a DNS server. If the DNS server is not connected to the same LAN the device must learn the IP address of the DNS server from the DHCP server during the process of obtaining its dynamically assigned IP address from the DHCP server.

DNS Servers

DNS servers are used to provide a network service that maps hostnames to IP addresses and IP addresses to hostnames (reverse DNS lookups). Anytime that you use a hostname to initiate an IP connection to a host, your PC must determine the IP address that is assigned to the hostname that you want to contact. For example, when you visit Cisco's website (http://www.cisco.com/) your PC sends a DNS query to a DNS server to discover the current IP address that can be used to contact Cisco's website.

For more information on DNS services visit the IETF RFC site (http://www.ietf.org/rfc.html) and look for RFCs about DNS. The Name Server LookUp tool (nslookup) is very useful for learning more about DNS. There are several excellent websites available about nslookup that you can find by searching for them.

Services and Servers Used By AutoInstall: Storage and Transmission of Configuration Files

TFTP is a protocol used to transfer files between devices on a network. A TFTP server is a device that uses TFTP to transfer files to devices. TFTP servers can be configured on UNIX servers, Microsoft Windows-based PCs and servers, and other platforms.

Tip If you do not have a TFTP server available you can configure a Cisco IOS-based router as a TFTP server using the **tftp-server file-system**:*filename* command. Refer to the *Configuring Basic File Transfer Services* guide http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/ffcprt2/fcf011.htm for more information on configuring your router as a TFTP server.

Cisco routers use TFTP to load the configuration files that are required for AutoInstall. You must have a TFTP server deployed in your network to provide file storage and file transmission services to the devices that will be using AutoInstall.

For more information on TFTP services visit the IETF RFC site (http://www.ietf.org/rfc.html) and look for RFCs about TFTP. There are several excellent websites available about TFTP that you can find by searching for them. Several freeware and shareware versions of TFTP servers for various operating systems and hardware platforms are available from the Internet.

The following caveats apply to the provisioning of TFTP servers for AutoInstall:

- Devices using AutoInstall over a LAN—If the TFTP server and the devices using AutoInstall are on different LAN segments, you must either configure the **ip helper-address** *address* command on all of the interfaces that will receive TFTP session initialization requests from the devices that are using AutoInstall.
- Devices using AutoInstall over a WAN—If the devices using AutoInstall are connected to a WAN, you must configure the **ip helper-address** *address* command on all of the interfaces that will receive TFTP session initialization requests from devices that are using AutoInstall.

ip helper-address

If the new device does not learn the IP address of the TFTP server via DHCP option 150, it will transmit the TFTP session initialization requests as network layer broadcasts using the IP destination broadcast address of 255.255.255.255. Routers block network layer broadcast datagrams which prevents the TFTP session initialization requests from reaching the TFTP server, and AutoInstall will fail. The solution to this problem is to use the **ip helper-address** *address* command. The **ip helper-address** *address* command changes the broadcast address of TFTP session initialization request from 255.255.255.255 to the address that is configured with the *address* argument. For example, the **ip helper-address 172.16.29.252** command will change IP destination broadcast address of 255.255.255.255 to 172.16.29.252.

Networking Devices Used by AutoInstall

These networking devices are used by AutoInstall:

- Device That Is Being Configured with AutoInstall, page 8
- Staging Router, page 8
- Intermediate Frame Relay-ATM Switching Device (Optional), page 9

Device That Is Being Configured with AutoInstall

A device that is being configured with AutoInstall can be any Cisco IOS-based router that supports AutoInstall and does not have a configuration file in its NVRAM.

Staging Router

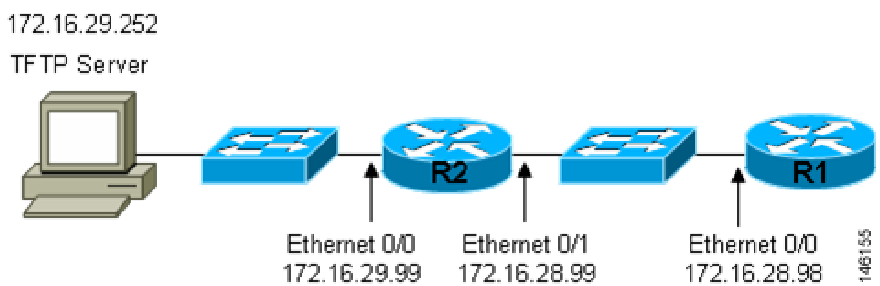
A staging router acts as an intermediary between the TFTP server (to which it must have IP connectivity) and a device that is being configured with AutoInstall when the new device and the TFTP server are connected to different networks. In Figure 3 R1 requires a staging router because it is connected to a different LAN segment than the TFTP server.

Staging routers are required in the following situations:

- Devices using AutoInstall over a LAN—If the TFTP and/or DHCP servers and the devices using AutoInstall are on different LAN segments you must use a staging router.

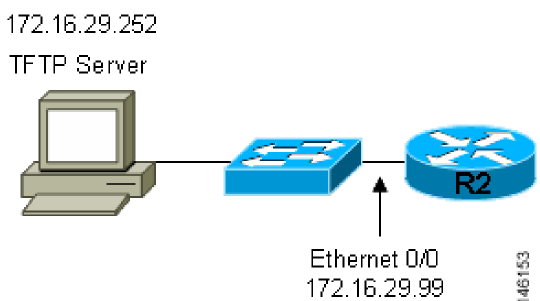
- Devices using AutoInstall over a WAN—If the devices using AutoInstall are connected to a WAN, you must configure the **ip helper-address** *address* command on all of the directly connected interfaces that will receive TFTP session initialization requests from the devices that are using AutoInstall.

Figure 3 Example of AutoInstall That Requires a Staging Router



Staging routers are not required when the new device that is being configured with AutoInstall is connected to the same LAN segment as the TFTP and DHCP servers. In Figure 4 R2 does not require a staging server to use AutoInstall because it is on the same LAN segment as the TFTP server.

Figure 4 Example of AutoInstall That Does Not Require a Staging Router



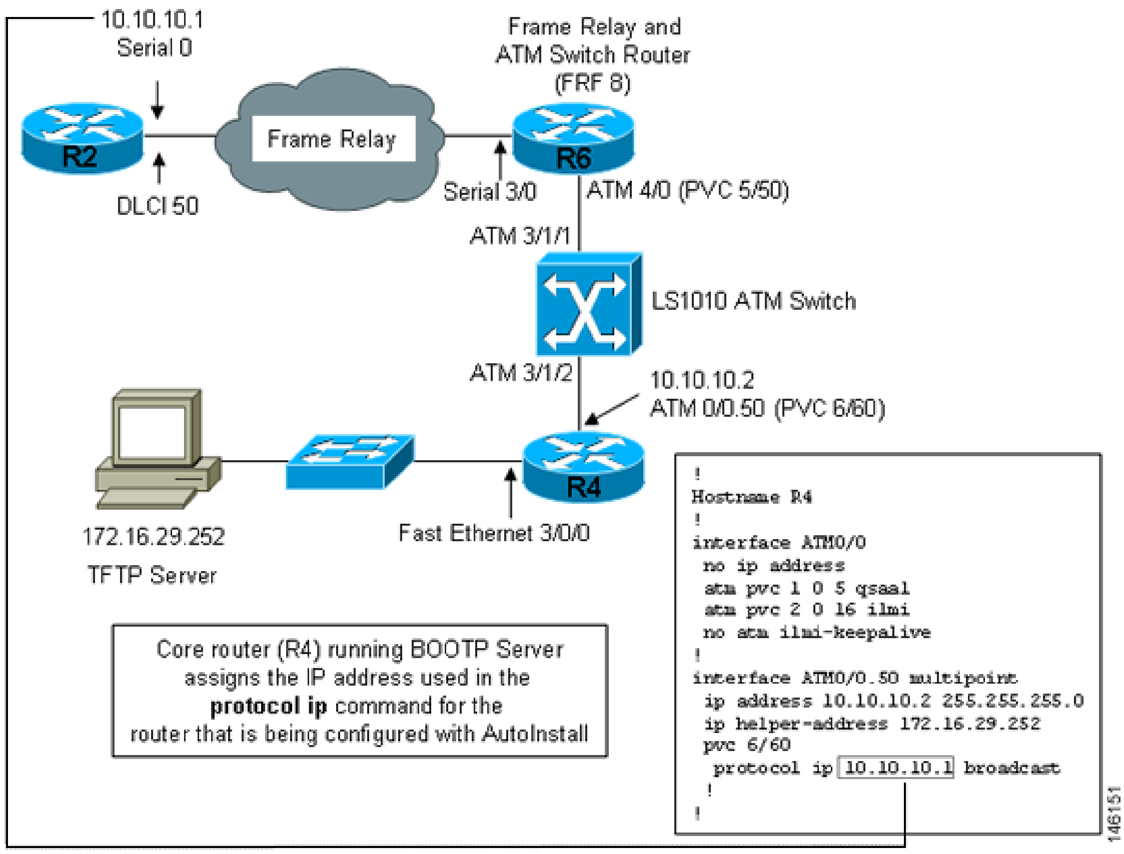
Intermediate Frame Relay-ATM Switching Device (Optional)

An intermediate Frame Relay-ATM switching device is one that can perform both routing and switching operations. Frame Relay-ATM switching devices are used to connect Frame Relay and ATM networks.

The AutoInstall over Frame Relay-ATM Interworking Connections feature modifies the AutoInstall process to use Frame Relay encapsulation defined by the IETF standard instead of the Frame Relay encapsulation defined by Cisco.

Figure 5 shows an example topology using AutoInstall over Frame Relay-ATM Interworking Connections. Router R6 does the Frame Relay to ATM Service Internetworking (FRF8) conversion for Frame Relay DLCI 50 to ATM VPI/VCI 5/50. The LS1010 switch routes the VPI/VCI combination used by R6 (5/50) to the VPI/VCI combination used by R4 (6/60).

Figure 5 Example Topology for AutoInstall over Frame Relay-ATM Interworking Connections



Configuration Files Used by AutoInstall

A configuration file executes predefined commands and settings that enable a device to function in a network. The type of configuration file you choose determines many aspects of how you set up the network for AutoInstall.

These types of files are used by AutoInstall:

- Network Configuration File, page 10
- Host-Specific Configuration File, page 11
- Default Configuration File (Optional), page 11

Network Configuration File

This is the first file that the AutoInstall process attempts to use. After the device has obtained an IP address it will try to discover its hostname by attempting to download a network configuration file that contains IP address to host name mappings.

If you want the device to learn its hostname from the network-confg file so that it can download a host-specific configuration file, you must add an entry for the device in the network-confg network configuration file. The syntax for the entry is **ip host** *hostname ip-address* where *hostname* is the name that you want the host to use and *ip-address* is the address that the host will receive from the IP address server. For example, if you want the new device to use the name Australia, and the IP address that was dynamically assigned the new device is 172.16.29.103, you need to create an entry in the network configuration file that contains the **ip host australia 172.16.29.103** command.

The file names used for the network configuration file are network-confg or cisconet.cfg. Routers running AutoInstall will try to load the network-confg from the TFTP server first. If the network-confg is not found on the TFTP server, the AutoInstall process will attempt to load the cisconet.cfg file. The cisconet.cfg filename was used by DOS-based TFTP servers that only supported the old 8.3 file naming convention. We recommend that you use the network-confg filename to avoid the delay that is created when AutoInstall has to timeout attempting to load the network-confg before it attempts to load the cisconet.cfg file.

If you using autoinstall to setup multiple devices you can create one network configuration file that contains an entry for each of the devices.

Host-Specific Configuration File

Host-specific configuration files are a full configuration for each new device. If you decide to use host-specific files, you must create a separate file for each new device that you are using AutoInstall to setup.

The filenames used for the host-specific configuration files are *name-confg* or *name.cfg* where the word name is replaced by the hostname of the router. For example, the filename for a router named hqrouter is hqrouter-confg or hqrouter.cfg.

Routers running AutoInstall will try to load the host-specific configuration filename using the format *name-confg* from the TFTP server first. If the *name-confg* file is not found on the TFTP server, the AutoInstall process will attempt to load the *name.cfg* file. The *name.cfg* file name format was used by DOS based TFTP servers that only supported the old 8.3 file naming convention. We recommend that you use the *name-confg* filename to avoid the delay that is created when AutoInstall has to timeout attempting to load the *name-confg* before it attempts to load the *name.cfg* file.

Tip If you use the *name.cfg* format for host-specific configuration files the filenames for hostnames that are longer than 8 characters must be truncated to the first eight characters. For example, the filename for a device with the hostname australia must be truncated to australi.cfg. When AutoInstall maps the IP address assigned to the new router to its hostname of australia in the network configuration file, AutoInstall will attempt to download a host-specific file with the name australi.cfg after it fails to load the host-specific filename austrailia-confg.

Tip Cisco recommends that you use the host-specific file option for setting up new devices to ensure that each new device is set up properly.

Default Configuration File (Optional)

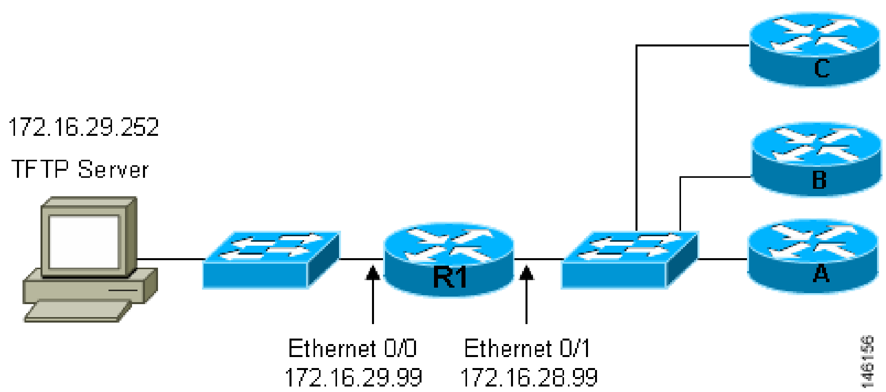
A default configuration file, which includes minimum configuration information allows you to telnet to the new device and configure it manually.

Tip If the new device has learned its hostname after it loaded the network configuration file the default configuration file is not used. You must use the host-specific file instead to configure features such as passwords for remote CLI sessions.

Figure 6 is an example of using the default configuration file to stage new routers for remote manual configuration. Routers A, B, and C are new routers that will be added to the network one at a time. You connect the first router and wait for it to load the default configuration file. The default configuration file must have enough information in it to allow the new router to communicate with the PC that you will be using to finish its configuration using a Telnet session. After the default configuration file is loaded on the new router, you can use Telnet to connect to the router to complete its configuration. You must assign a new, unique IP address to its interfaces so that the default configuration file can be used for configuring the next router.

Caution Failure to change the IP addresses in the router that you are configuring remotely with Telnet will result in duplicate IP addresses on the LAN when the next router loads the default configuration file. In this situation you will not be able to use Telnet to connect to either router. You must disconnect one of the routers before you can resolve this problem.

Figure 6 Example of Using the Default Configuration File To Stage Routers For Remote Manual Configuration



Tip You must include the commands for configuring passwords for remote Telnet access and access to privileged EXEC mode if you are going to access the routers remotely to complete their configurations save their configuration files to NVRAM.

The filenames used for the default network configuration file are router-confg or router.cfg. Routers running AutoInstall will try to load the router-confg from the TFTP server first. If the router-confg is not found on the TFTP server the AutoInstall process will attempt to load the router.cfg file. The router.cfg file name was used by DOS-based TFTP servers that only supported the old 8.3 file naming convention. We recommend that you use the router-confg filename to avoid the delay that is created when AutoInstall has to timeout while attempting to load the router-confg before it attempts to load the router.cfg file.

If you are using AutoInstall to configure LAN-attached devices, you can specify a different default boot filename in DHCP Option 067.

Configuration Options for AutoInstall

You can provision your network to support AutoInstall using several different combinations of devices and services. For example:

- You can provision all of the services required for AutoInstall (except dynamic IP address assignment using SLARP or BOOTP that must be preformed by a Cisco router) on one network server, or you can provision each service on a different network server.
- You can provision the DHCP service on a Cisco router.
- The device using AutoInstall can determine its IP address from a DNS server, or you can use one of the AutoInstall network configuration files (network-confg or cisconet.cfg) that contain the **ip host** *hostname ip-address* commands.
- You can use provision AutoInstall to load a full configuration or a partial configuration onto a device that is using AutoInstall.

This module focuses on some of the most common methods for provisioning AutoInstall. Refer to the "How to Use AutoInstall to Remotely Configure Cisco Networking Devices" section on page 15 for information on the most common methods for provisioning AutoInstall.

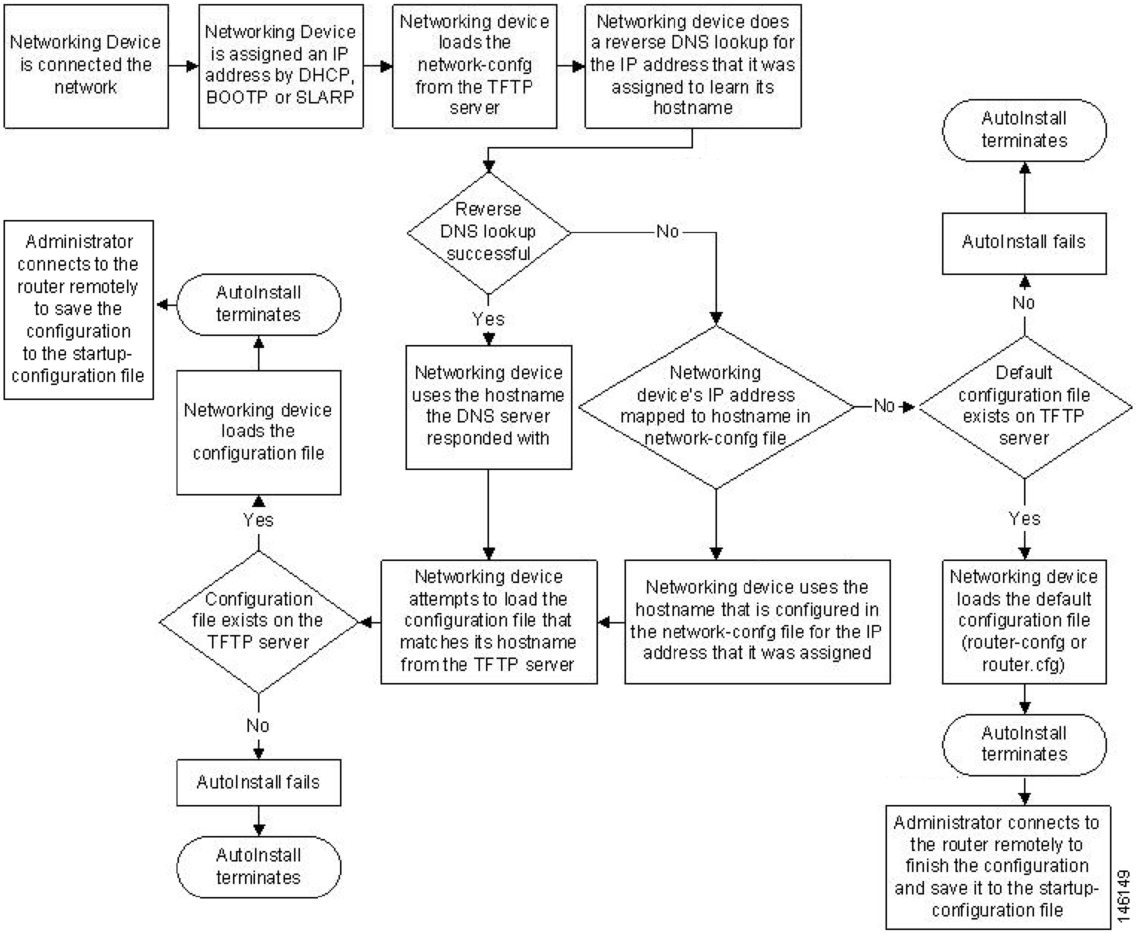
The AutoInstall Process

The AutoInstall process begins when a networking device that does not have any files in its NVRAM is connected to the network.

Timesaver You can decrease the time that the AutoInstall process takes to complete by only connecting the interface on the networking device that you want to use for AutoInstall until the AutoInstall process has finished. For example, if you want the networking device to perform AutoInstall over a WAN interface and you connect its LAN interfaces and its WAN interfaces the networking device will attempt to perform AutoInstall over the LAN interfaces before it attempts to use the WAN interfaces. Leaving the LAN interfaces disconnected until the AutoInstall process is finished causes the networking device to initiate the AutoInstall process over its WAN interface immediately.

Figure 7 shows the basic flow of the AutoInstall process.

Figure 7 AutoInstall Process Flowchart



Benefits of Using AutoInstall to Remotely Configure a Cisco Networking Device

AutoInstall facilitates the deployment of Cisco routers by allowing you to manage the setup procedure for routers from a central location. The person responsible for physically installing the router does not require specific networking skills. The ability to physically install the router, connect the power and networking cables, and power it on are the only skills required by the installer. The configuration files are stored and managed on a central TFTP server. By using AutoInstall one skilled network technician based at a central site can manage the deployment of several routers in a short period of time.

Two enhancements to AutoInstall:

- AutoInstall Using DHCP for LAN Interfaces
- AutoInstall over Frame Relay-ATM Interworking Connections

AutoInstall Using DHCP for LAN Interfaces

The AutoInstall Using DHCP for LAN Interfaces feature enhances the benefits of AutoInstall by replacing the use of the Bootstrap Protocol (BOOTP) with the use of the Dynamic Host Configuration Protocol (DHCP) for Cisco IOS AutoInstall over LAN interfaces (specifically Ethernet, Token Ring, and FDDI interfaces).

DHCP (defined in RFC 2131) is an extension of the functionality provided by the BOOTP (defined in RFC 951). DHCP provides the framework for passing configuration information to hosts on a TCP/IP network. DHCP adds the capability of automatic allocation of reusable network addresses and additional configuration options. In Cisco IOS Release 12.1(5)T, and later releases, the IP address procurement phase of the AutoInstall process is now accomplished using DHCP for Ethernet, Token Ring, and FDDI interfaces. Prior to this release, IP addresses for LAN interfaces were obtained using BOOTP or RARP during the AutoInstall process. Additionally, this feature allows for the uploading of configuration files using unicast TFTP.

AutoInstall over Frame Relay-ATM Interworking Connections

The AutoInstall over Frame Relay-ATM Interworking Connections feature further enhances the benefits of AutoInstall by allowing you to use a router with an ATM interface as a BOOTP server for new routers being connected at remote locations.

How to Use AutoInstall to Remotely Configure Cisco Networking Devices

This section describes the how to prepare a router for AutoInstall, and how to use AutoInstall with Frame Relay to ATM Service Internetworking. Additional examples for using AutoInstall for new routers connected to LANs, HDLC WANs, and Frame Relay networks that do not use Frame Relay to ATM Service Internetworking, are provided in the "Configuration Examples for Using AutoInstall to Remotely Configure Cisco Networking Devices" section on page 31.

In most cases you need to configure a staging router through which a new device running AutoInstall sends TFTP, BOOTP, and DNS requests.

Tip In all cases, you must verify and save the configuration on the networking device after the AutoInstall process is complete. If you do not save the configuration, you must repeat the entire process.

Disabling the SDM Default Configuration File

Perform this task if SDM was pre installed on your device and you want to use AutoInstall to configure the device instead. SDM remains on the device.

SUMMARY STEPS

1. Connect the console cable from the console port on the device to the serial port on the PC.
2. Connect the power supply to the device, plug the power supply into a power outlet, and turn on the device.
3. Connect to the device using a terminal emulation program.
4. **enable**
5. **erase startup-config**

6. **reload**

DETAILED STEPS

Step 1 Connect the console cable, shipped with your device, from the console port on the device to a serial port on your PC. Refer to the hardware installation guide for the device for instructions.

Step 2 Connect the power supply to the device, plug the power supply into a power outlet, and turn on the device. Refer to the quick start guide for the device for instructions.

Step 3 Use Hyperterminal or a similar terminal emulation program on your PC, with the following terminal emulation settings, to connect to the device:

- 9600 baud
- 8 data bits, no parity, 1 stop bit
- No flow control

Step 4 **enable**

Enter privileged EXEC mode.

enable

```
Router> enable
Router#
```

Step 5 **erase startup-config**

Erases the existing configuration in NVRAM.

```
Router# erase startup-config
```

Step 6 **reload**

Initiates the reload process. The router will initiate the AutoInstall process after it finishes the reload process.

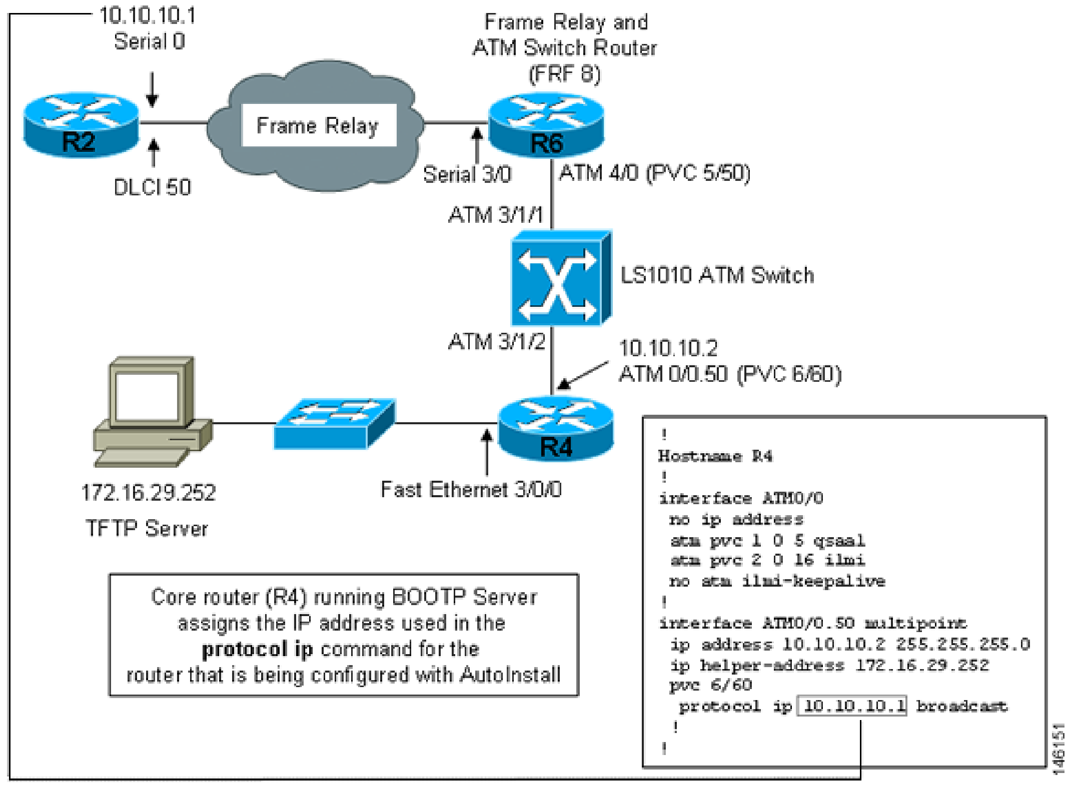
```
Router# reload
```

Using AutoInstall with Frame Relay to ATM Service Internetworking

Refer to figure 8 for the sample network used in this task. Perform this task to configure routers R6, R4, and the LS1010 ATM switch so that AutoInstall can be used with Frame Relay to ATM Service Internetworking (FRF8) to setup router R2.

Tip The IP address that will be assigned to Serial 0 on R2 (10.10.10.1/24) during and after the AutoInstall process and the IP address that is assigned to ATM 0/0.50 on R4 (10.10.10.2/24) are on the same subnet (10.10.10.0/24). Using IP addresses on the same subnet is required because the interfaces on R6 and the LS10101 switch are switching the IP packets between R2 and R4 at Layer 2.

Figure 8 Example Topology for AutoInstall over Frame Relay/ATM Interworking Connections



This sections contains the following tasks:

- Configuring R6 for Frame Relay to ATM Service Internetworking, page 17
- Verifying Frame Relay to ATM Service Internetworking on R6, page 21
- Configuring R4 for Frame Relay to ATM Service Internetworking, page 21
- Configuring IP Routing R4, page 24
- Configuring the LS1010 Switch, page 25
- Creating the Configuration File for R2, page 27
- Verifying AutoInstall with Frame Relay to ATM Service Internetworking, page 28

Configuring R6 for Frame Relay to ATM Service Internetworking

Router R6 does the Frame Relay to ATM Service Internetworking (FRF8) conversion for Frame Relay DLCI 50 to ATM VPI/VCI 5/50.

Note The serial interface and the ATM interface on R6 that are used for ATM Service Internetworking (FRF8) do not have IP addresses because they are used as Layer 2 switching interfaces in this configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **hostname** *hostname*
4. **interface serial** *interface-number*
5. **no ip address**
6. **encapsulation frame-relay ietf**
7. **frame-relay interface-dlci** *dlci* **switched**
8. **frame-relay lmi-type ansi**
9. **frame-relay intf-type dce**
10. **exit**
11. **interface atm** *interface-number*
12. **no ip address**
13. **atm pvc** *number 0 5* **qsaal**
14. **atm pvc** *number 0 16* **ilmi**
15. **no atm ilmi-keepalive**
16. **pvc** *vpi vci*
17. **encapsulation aal5mux fr-atm-srv**
18. **exit**
19. **exit**
20. **connect** *name* **serial** *interface-number dlci* **atm** *interface-number vpi/vci* **service-interworking**
21. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**hostname** *hostname* **Example:** `Router(config)# hostname R6`	In this example, the hostname is configured as R6.
Step 4	**interface serial** *interface-number* **Example:** `R6(config)# interface serial 3/0`	Specifies the serial interface that connects to the router that is being setup with AutoInstall.

	Command or Action	Purpose
Step 5	**no ip address** **Example:** `R6(config-if)# no ip address`	Removes an existing IP address. **Note** This interface is used as a layer 2 switch interface in this configuration. It is not an IP layer 3 endpoint. Therefore it does not require an IP address.
Step 6	**encapsulation frame-relay ietf** **Example:** `R6(config-if)# encapsulation frame-relay IETF`	Enables and specifies the Frame Relay encapsulation method. **Note** Only the Frame Relay commands and keywords required for this task are described in this task. For more information on the other Frame Relay commands and keywords, refer to the *Cisco IOS Wide-Area Networking Command Reference*,
Step 7	**frame-relay interface-dlci** *dlci* **switched** **Example:** `R6(config-if)# frame-relay interface-dlci 50 switched`	Specifies that the Frame Relay data-link connection identifier (DLCI) is switched.
Step 8	**frame-relay lmi-type ansi** **Example:** `R6(config-if)# frame-relay lmi-type ansi`	Specifies that the router should use Annex D defined by American National Standards Institute (ANSI) standard T1.617 as the LMI type.
Step 9	**frame-relay intf-type dce** **Example:** `R6(config-if)# frame-relay intf-type dce`	Specifies that the router functions as a switch connected to a router.
Step 10	**exit** **Example:** `R6(config-if)# exit`	Returns to global configuration mode.
Step 11	**interface atm** *interface-number* **Example:** `R6(config)# interface ATM4/0`	Species the ATM interface and enters interface configuration mode. **Note** Only the ATM commands and keywords required for this task are described in this task. For more information on the other Frame Relay commands and keywords refer to the *Cisco IOS Asynchronous Transfer Mode Command Reference*.
Step 12	**no ip address** **Example:** `R6(config-if)# no ip address`	Removes an existing IP address. **Note** This interface is used as a layer 2 switch interface in this configuration. It is not an IP layer 3 endpoint. Therefore it does not require an IP address.
Step 13	**atm pvc number 0 5 qsaal** **Example:** `R6(config-if)# atm pvc 1 0 5 qsaal`	Configures a PVC for QSAAL1 signaling.

	Command or Action	Purpose
Step 14	**atm pvc number 0 16 ilmi** **Example:** `R6(config-if)# atm pvc 2 0 16 ilmi`	Configures a PVC for ILMI signaling.
Step 15	**no atm ilmi-keepalive** **Example:** `R6(config-if)# no atm ilmi-keepalive`	Disables ATM ILMI keep alives.
Step 16	**pvc** *vpi/vci* **Example:** `R6(config-if)# pvc 5/50`	Configures the PVC. When configuring PVCs, configure the lowest available VPI and VCI numbers first. **Note** VCIs 0 to 31 on all VPIs are reserved.
Step 17	**encapsulation aal5mux fr-atm-srv** **Example:** `R6(config-if-atm-vc)# encapsulation aal5mux fr-atm-srv`	Enables the Frame Relay and ATM internetworking service.
Step 18	**exit** **Example:** `R6(config-if-atm-vc)# exit`	Exits PVC configuration mode and returns to interface configuration mode.
Step 19	**exit** **Example:** `R6(config-if)# exit`	Returns to global configuration mode.
Step 20	**connect** *name* **serial** *slot/port dlci* **atm** *slot/port vpi/vci* **service-interworking** **Example:** `R6(config)# connect r2 serial3/0 50 ATM4/0 5/50 service-interworking`	Creates the connection between the Frame Relay DLCI and the ATM PVC for the Frame Relay and ATM internetworking service.
Step 21	**end** **Example:** `R6(config)# end`	Returns to privileged EXEC mode.

Examples

The following example shows how to configure R6 for Frame Relay to ATM Service Internetworking (FRF8).

```
!
hostname R6
!
interface Serial3/0
 no ip address
 encapsulation frame-relay IETF
 frame-relay interface-dlci 50 switched
 frame-relay lmi-type ansi
```

```
 frame-relay intf-type dce
!
interface ATM4/0
 atm pvc 1 0 5 qsaal
 atm pvc 2 0 16 ilmi
 no atm ilmi-keepalive
 pvc 5/50
  encapsulation aal5mux fr-atm-srv
!
connect r2 serial3/0 50 atm4/0 5/50 service-interworking
!
```

Verifying Frame Relay to ATM Service Internetworking on R6

In this example the output of the **show connection name r2** command indicates that the Service Interworking Connection is up.

```
R6# show connection name r2

FR/ATM Service Interworking Connection: r2
  Status    - UP
  Segment 1 - Serial3/0 DLCI 50
  Segment 2 - ATM4/0 VPI 5 VCI 50
Interworking Parameters -
    service translation
    efci-bit 0
    de-bit map-clp
    clp-bit map-de
```

Configuring R4 for Frame Relay to ATM Service Internetworking

R4 is one of the endpoints for Frame Relay to ATM Service Internetworking in this task. R2 is the other endpoint. R4 is not directly connected to the Frame Relay network. Therefore R4 requires only the ATM commands to act as the endpoint for Frame Relay to ATM Service Internetworking.

R4 is the core router that connects to the LAN with the TFTP server. R4 is the BOOTP server that will provide the IP address required for R2 (10.10.10.1/24) when R2 runs AutoInstall.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *hostname*
4. **interface ethernet** *interface-number*
5. **ip address** *ip-address mask*
6. **interface atm** *interface-number*
7. **no ip address**
8. **atm pvc** *number 0 5* **qsaal**
9. **atm pvc** *number 0 16* **ilmi**
10. **no atm ilmi-keepalive**
11. **interface atm** *interface-number.subinterface-number* **multipoint**
12. **ip address** *ip-address mask*

13. **ip helper-address** *ip-address*
14. **pvc** *vpi/vci*
15. **protocol ip** *ip-address* **broadcast**
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`hostname`** *`hostname`* **Example:** `Router(config)# hostname R4`	In this example the hostname is configured as R4.
Step 4	**`interface ethernet`** *`module/slot/port`* **Example:** `R4(config)# interface ethernet 3/0/0`	Species the Ethernet interface and enters interface configuration mode.
Step 5	**`ip address`** *`ip-address mask`* **Example:** `R4(config-if)# ip address 172.16.29.97 255.255.255.0`	Specifies the IP address for the interface.
Step 6	**`interface atm`** *`interface-number`* **Example:** `R4(config)# interface atm0/0`	Species the ATM interface and enters interface configuration mode. **Note** Only the ATM commands and keywords required for this task are described in this task. For more information on the other Frame Relay commands and keywords, refer to the *Cisco IOS Asynchronous Transfer Mode Command Reference*.
Step 7	**`no ip address`** **Example:** `R4(config-if)# no ip address`	The main ATM interface does not require an IP address in this configuration. The IP address is assigned to the multipoint subinterface in Step 9.
Step 8	**`atm pvc number 0 5 qsaal`** **Example:** `R4(config-if)# atm pvc 1 0 5 qsaal`	Configures a PVC for QSAAL1 signaling.

	Command or Action	Purpose
Step 9	**atm pvc number 0 16 ilmi** **Example:** `R4(config-if)# atm pvc 2 0 16 ilmi`	Configures a PVC for ILMI signaling.
Step 10	**no atm ilmi-keepalive** **Example:** `R4(config-if)# no atm ilmi-keepalive`	Disables ATM ILMI keep alives.
Step 11	**interface atm** *slot/port.subinterface-number* **multipoint** **Example:** `R4(config-if)# interface atm0/0.50 multipoint`	Creates the ATM multipoint virtual sub-interface and enters sub-interface configuration mode.
Step 12	**ip address** *ip-address mask* **Example:** `R4(config-subif)# ip address 10.10.10.2 255.255.255.0`	Specifies the IP address for the sub-interface.
Step 13	**ip helper-address** *ip-address* **Example:** `R4(config-subif)# ip helper-address 172.16.29.252`	Specifies the IP address of the TFTP server. This IP address is used to replace the 255.255.255.255 IP destination broadcast address that R2 will use when it attempts to establish a connection to the TFTP server.
Step 14	**pvc** *vpi/vci* **Example:** `R4(config-if-atm-vc)# pvc 6/60`	Configures the PVC. When configuring PVCs, configure the lowest available VPI and VCI numbers first. **Note** VCIs 0 to 31 on all VPIs are reserved.
Step 15	**protocol ip** *ip-address* **broadcast** **Example:** `R4(config-if-atm-vc)# protocol ip 10.10.10.1 broadcast`	Specifies the IP address of the device at the other end of this PVC. In this example the device is R2. For this example this is the IP address that will be assigned by the BOOTP server on R4 to R2 during the AutoInstall process.
Step 16	**end** **Example:** `R4(config-if-atm-vc)# end`	Returns to privileged EXEC mode.

Examples

The following example configures R4 as the core router for AutoInstall using Frame Relay to ATM Service Internetworking (FRF8).

```
!
hostname R4
!
interface FastEthernet3/0/0
 ip address 172.16.29.97 255.255.255.0
!
```

```
interface ATM0/0
 no ip address
 atm pvc 1 0 5 qsaal
 atm pvc 2 0 16 ilmi
 no atm ilmi-keepalive
!
interface ATM0/0.50 multipoint
 ip address 10.10.10.2 255.255.255.0
 ip helper-address 172.16.29.252
 pvc 6/60
  protocol ip 10.10.10.1 broadcast
 !
!
```

Configuring IP Routing R4

In order for R4 to be able to forward IP traffic between network 172.16.29.0 and R2 after the AutoInstall process is complete, R4 needs to have IP routing configured.

Note The configuration file for R2 provided in the "Creating the Configuration File for R2" section on page 27 includes the IP routing commands required to establish IP routing connectivity for R2 using RIP Version 2.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router rip**
4. **version** *version*
5. **network** *ip-network*
6. Repeat step 5 for the other IP networks.
7. **no auto-summary**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `R4> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `R4# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router rip** **Example:** `R4(config)# router rip`	Enables RIP routing on R4. **Note** Only the RIP commands and keywords required for this task are described in this task. For more information on the other RIP commands and keywords, refer to the *Cisco IOS Routing Protocols Command Reference*.
Step 4	**version** *version* **Example:** `R4(config-router)# version 2`	Specifies the version of RIP that the router will use.
Step 5	**network** *ip-network* **Example:** `R4(config-router)# network 172.16.0.0`	Specifies the IP networks that RIP will provide routing services for.
Step 6	Repeat Step 5 for the other IP networks. **Example:** `R4(config-router)# network 10.0.0.0`	—
Step 7	**no auto-summary** **Example:** `R4(config-router)# no auto-summary`	Disables the default RIP V2 behavior of summarizing IP subnets in the routing advertisements.
Step 8	**end** **Example:** `R4(config-router)# end`	Returns to privileged EXEC mode.

Examples

The following example shows how to configure IP routing on R4.

```
!
router rip
 version 2
 network 10.0.0.0
 network 172.16.0.0
 no auto-summary
!
```

Configuring the LS1010 Switch

This task describes how to configure an LS1010 switch to route the PVCs between R6 and R4. R6 is connected to ATM 3/1/1 on the LS1010 switch. R4 is connected to ATM 3/1/2 on the LS1010 switch.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface atm** *module/slot/port*
4. **atm pvc** *vpi vci* **interface atm** *interface-number vpi vci*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *module/slot/port* **Example:** `Switch(config)# interface ATM3/1/2`	Species the ATM interface and enters interface configuration mode. **Note** Only the LS1010 ATM commands and keywords required for this task are described in this task. For more information on the other ATM commands and keywords available on the LS1010, refer to the *Lightstream 1010 ATM Switch Documents*, Release 12.1(26)E3. (http://www.cisco.com/univercd/cc/td/doc/product/atm/ls1010s/12_1/26_e3/index.htm)
Step 4	**atm pvc** *vpi vci* **interface atm** *interface-number vpi vci* **Example:** `Switch(config-if)# atm pvc 6 60 interface ATM3/1/1 5 50`	Configures a static PVC route In this example a route for the PVC from R6 (5/50) to R4 (6/60) is configured.
Step 5	**end** **Example:** `Switch(config-if)# end`	Returns to privileged EXEC mode.

Examples

The following example shows how to configure the LS1010 ATM switch to route the PVCs between R6 and R4.

```
!
atm address 47.0091.8100.0000.0010.11b9.6101.0010.11b9.6101.00
atm router pnni
 no aesa embedded-number left-justified
 node 1 level 56 lowest
  redistribute atm-static
!
interface ATM2/0/0
 no ip address
```

```
 no ip directed-broadcast
 atm maxvp-number 0
!
interface ATM3/1/0
 no ip address
 no ip directed-broadcast
 no atm ilmi-keepalive
!
interface ATM3/1/1
 no ip address
 no ip directed-broadcast
 no atm ilmi-keepalive
!
interface ATM3/1/2
 no ip address
 no ip directed-broadcast
 no atm ilmi-keepalive
 atm pvc 6 60 interface ATM3/1/1 5 50
!
interface ATM3/1/3
 no ip address
 no ip directed-broadcast
 no atm ilmi-keepalive
!
```

Creating the Configuration File for R2

This section provides the content for the configuration file for R2.

SUMMARY STEPS

1. Create the configuration file for R2 using the information provided.
2. Store the configuration file on the TFTP server with the name r2-confg.

DETAILED STEPS

Step 1 Create the following configuration file for R2

```
!
hostname R2
!
!
enable secret 7gD2A0
!
interface Ethernet0
 no ip address
 shutdown
!
interface Serial0
 ip address 10.10.10.1 255.255.255.0
 encapsulation frame-relay IETF
 frame-relay map ip 10.10.10.2 50 broadcast
 frame-relay interface-dlci 50
 frame-relay lmi-type ansi
!
interface Serial1
 no ip address
 shutdown
!
```

```
!
router rip
 version 2
 network 10.0.0.0
 no auto-summary
!
ip http server
ip classless
!
line vty 0 4
 password 87F3c0m
 login
!
end
```

Step 2 Store the configuration file on the TFTP server with the name r2-confg

Verifying AutoInstall with Frame Relay to ATM Service Internetworking

This task verifies the AutoInstall with Frame Relay to ATM Service Internetworking configuration by setting up R2, as shown in Figure 8, using AutoInstall.

Prerequisites

The following prerequisites must be met before you can perform this task:

- You must have a TFTP server on the network with the IP address that you specified on R4 with the **ip helper-address** *ip-address* command.
- You must have a configuration file for R2 named r2-confg on the TFTP server.
- You must have a network configuration named network-confg file with the **ip host r2 10.10.10.1** command in it on the TFTP server.
- You must have configured R6, R4 and the LS1010 ATM switch (or a functional equivalent of the ATM switch) following the instructions provided in the previous tasks in this section.
- R2 must not have a configuration file in NVRAM.

Step 1 Connect a console terminal to R2.

Use Hyperterminal or a similar terminal emulation program on your PC, with the following terminal emulation settings, to connect to the device:

- 9600 baud
- 8 data bits, no parity, 1 stop bit
- No flow control

Step 2 Power cycle, or power on R2.

Step 3 When the prompt to enter the initial configuration dialog appears, answer no.

```
Would you like to enter the initial configuration dialog? [yes/no]: no
```

Step 4 When the prompt to terminate AutoInstall appears answer no.

```
Would you like to terminate autoinstall? [yes]: no
```

AutoInstall will start.

```
Please Wait. Autoinstall being attempted over Serial0 !!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Step 5 The AutoInstall process can take several minutes to complete. Do not press any keys on R2's terminal session until AutoInstall has completed.

This display output is from a successful Auto Installation process.

Note You can ignore the "%PARSER-4-BADCFG: Unexpected end of configuration file" error message. This problem does not adversely affect the AutoInstall process.

Note The last two lines with the %SYS-5-CONFIG_I messages indicate the network-confg and r2-confg files have been received successfully.

```
Press RETURN to get started!

*Mar  1 00:00:11.155: %LINK-3-UPDOWN: Interface Ethernet0, changed state to up
*Mar  1 00:00:11.159: %LINK-3-UPDOWN: Interface Serial0, changed state to up
*Mar  1 00:00:11.527: %LINK-3-UPDOWN: Interface Serial1, changed state to down
*Mar  1 00:00:12.271: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0,changed
state to up
```

```
*Mar  1 00:00:29.487: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed
state to down
*Mar  1 00:00:32.347: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed
state to up
*Mar  1 00:00:40.355: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed
state to down
*Mar  1 00:00:45.551: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed
state to up
*Mar  1 00:01:58.499: %IP-5-WEBINST_KILL: Terminating DNS process
*Mar  1 00:02:00.035: %LINK-5-CHANGED: Interface Ethernet0, changed state to
administratively down
*Mar  1 00:02:00.039: %LINK-5-CHANGED: Interface Serial1, changed state to
administratively down
*Mar  1 00:02:01.039: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed
state to down
*Mar  1 00:02:50.635: %SYS-5-RESTART: System restarted --
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-IS-L), Version 12.3(13a), RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Tue 26-Apr-05 12:52 by ssearch
*Mar  1 00:02:50.643: %SNMP-5-COLDSTART: SNMP agent on host Router is undergoing
 a cold start
*Mar  1 00:03:54.759: %PARSER-4-BADCFG: Unexpected end of configuration file.

*Mar  1 00:03:54.763: %SYS-5-CONFIG_I: Configured from tftp://172.16.29.252/network-confg
by console

*Mar  1 00:04:12.747: %SYS-5-CONFIG_I: Configured from tftp://172.16.29.252/r2-confg by
console
```

If you have logging enabled on your TFTP server the log should contain messages similar to the following text:

```
Sent network-confg to (10.10.10.1), 76 bytes
Sent r2-confg to (10.10.10.1),687 bytes
```

Step 6 Copy the running configuration to the startup-configuration with the **copy running-configuration startup-configuration** command.

Troubleshooting

If after approximately five minutes you do not see the %SYS-5-CONFIG_I messages and R2 has a factory default prompt of Router>, the AutoInstall process failed.

Step 1 Look for error messages on the TFTP server indicating that the files were not found. A very common mistake is that the .txt extension was added to the r2-confg file (r2-confg.txt) by your text editor. Your operating system might be hiding the extension for known file types when you browse the TFTP root directory. Disable the **Hide file extensions for known file types** option.

Tip You can stop most text editors from adding the filename extension by saving the file with double quotes (“filename“) around the filename. For example, saving the file as “r2-confg” should force the text editor to only use r2-confg.

Step 2 Test the connectivity in your network by configuring R2 with the configuration file that you created. You can copy the configuration for R2 to R2 by pasting it into the console terminal session.

After you have copied the configuration to R2, try to ping 10.10.10.2. If this fails, you have a problem between R2 and R4. Verify the cabling, the status of the interfaces, and the configurations on the routers.

If R2 can ping 10.10.10.2, try pinging the TFTP server (172.16.29.252) from R2. If this fails, you have a configuration problem somewhere between R4 and the TFTP server. Verify the cabling, the status of the interfaces, and the configurations on the routers. Verify the IP address and IP default gateway on the TFTP server.

Tip The IP default gateway on the TFTP server should be 172.16.29.97 (the local Ethernet interface on R4).

If R2 can ping the TFTP server (172.16.29.252), you probably have a problem with the TFTP server itself. A common mistake with TFTP servers is that they are configured to receive files but not to send them. Another common mistake on UNIX-based TFTP servers is that the files do not have the correct permissions. On a UNIX TFTP server the files should have permissions set to rw-rw-rw.

Step 3 If the IP connectivity appears to be working and the TFTP server is configured correctly, verify that you entered the **ip helper-address** *ip-address* command on R4 correctly.

Configuration Examples for Using AutoInstall to Remotely Configure Cisco Networking Devices

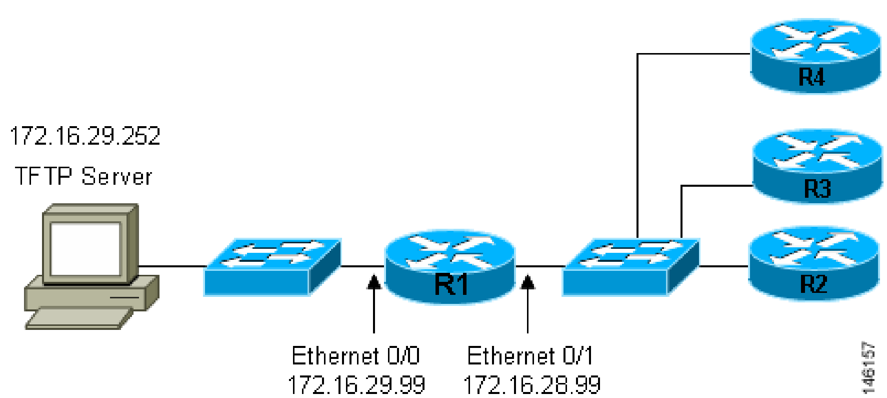
This section provides the following configuration examples:

- Using AutoInstall to Set Up Devices Connected to LANs: Example, page 31
- Using AutoInstall to Set Up Devices Connected to WANs: Example, page 43

Using AutoInstall to Set Up Devices Connected to LANs: Example

This task uses the network in Figure 9. This task will show how to use AutoInstall to setup routers R2, R3, and R4. Router R1 is the DHCP server that will be used to assign the IP address for Ethernet 0 on the new routers during the AutoInstall process.

Figure 9 Network Topology for Assigning AutoInstall Configuration Files For Specific Devices



Every DHCP client has a unique DHCP client identifier. The DHCP client identifier is used by DHCP servers to keep track of IP address leases and for configuring IP address reservations. You need to know the DHCP client identifier for each of the networking devices that you want to configure with AutoInstall so that you can configure the DHCP IP address reservations which will ensure that each device is provided with the correct IP address, and subsequently its unique configuration file. You can determine the DHCP client identifier manually or automatically.

To use AutoInstall to setup routers R2, R3, and R4, perform following tasks:

- Determining the Value for the DHCP Client Identifier Manually, page 32
- Determining the Value for the DHCP Client Identifier Automatically, page 35
- Creating a Private DHCP Pool for Each of The Routers, page 38
- Creating Configuration Files for Each Router, page 39
- Creating the network-confg file, page 40
- Setting Up the Routers with AutoInstall, page 40
- Saving the Configuration Files on The Routers, page 42
- Removing the Private DHCP Address Pools from R1, page 43

Determining the Value for the DHCP Client Identifier Manually

If you want to determine the value for the client identifiers automatically, you do not need to perform this task. Proceed to the "Determining the Value for the DHCP Client Identifier Automatically" section on page 35.

Tip If you are using AutoInstall to configure networking devices that are running an IOS release other than 12.4(1) or newer the DHCP client identifier might use a different format. In this case use the process explained in the "Determining the Value for the DHCP Client Identifier Automatically" section on page 35.

You must know the MAC address of the Ethernet interface that will be used to connect the router to the LAN during the AutoInstall process to determine the client identifier manually. This requires connecting a terminal to the router, and powering it on, so that you can enter the **show interface** *interface-type interface-number* command.

The client-identifier looks like this:

0063.6973.636f.2d30.3030.362e.3533.6237.2e38.6537.312d.4661.332f.30

The format is *nullcisco-0006.53b7.8e71-fa3/0* where *0006.53b7.8e71* is the MAC address and *fa3/0* is the short interface name for the interface that the IP address request is made for.

The values for the short-if-name field can be obtained from an SNMP workstation with the Cisco MIBs installed. This is an example of how to map ifIndex to an interface on Cisco IOS:

```
snmpwalk -c public ponch ifName
IF-MIB::ifName.1 = STRING: AT2/0
IF-MIB::ifName.2 = STRING: Et0/0
IF-MIB::ifName.3 = STRING: Se0/0
IF-MIB::ifName.4 = STRING: BR0/0
```

Use the **show interface** *interface-type interface-number* command to display the information and statistics for a FastEthernet interface.

```
R6> show interface fastethernet 3/0
FastEthernet3/0 is up, line protocol is up
  Hardware is AmdFE, address is 0006.53b7.8e71 (bia 0006.53b7.8e71)
.
.
.
R6>
```

The MAC address for FastEthernet 3/0 on R6 is 0006.53b7.8e71. The format of the client identifier for this interface is nullcisco-0006.53b7.8e71-fa3/0.

Note The short interface name for FastEthernet interfaces is fa.

Table 1 shows the values for converting characters to their hexadecimal equivalents. The last row in Table 2 shows the client identifier for FastEthernet 3/0 on R6 (nullcisco-0006.53b7.8e71-fa3/0).

Table 1 Hexadecimal to Character Conversion Chart

Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char
00	NUL	1a	SUB	34	4	4e	N	68	h
01	SOH	1b	ESC	35	5	4f	O	69	I
02	STX	1c	FS	36	6	50	P	6a	j
03	ETX	1d	GS	37	7	51	Q	6b	k
04	EOT	1e	RS	38	8	52	R	6c	l
05	ENQ	1f	US	39	9	53	S	6d	m
06	ACK	20		3a	:	54	T	6e	n
07	BEL	21	!	3b	;	55	U	6f	o
08	BS	22	"	3c	<	56	V	70	p
09	TAB	23	#	3d	=	57	W	71	q
0A	LF	24	$	3e	>	58	X	72	r
0B	VT	25	%	3f	?	59	Y	73	s
0C	FF	26	&	40	@	5a	Z	74	t
0D	CR	27	‘	41	A	5b	[	75	u
0E	SO	28	(	42	B	5c	\	76	v
0F	SI	29	)	43	C	5d	]	77	w
10	DLE	2a	*	44	D	5e	^	78	x
11	DC1	2b	+	45	E	5f	_	79	y
12	DC2	2c	,	46	F	60	`	7a	z
13	DC3	2d	-	47	G	61	a	7b	{
14	DC4	2e	.	48	H	62	b	7c	|
15	NAK	2f	/	49	I	63	c	7D	}
16	SYN	30	0	4a	J	64	d	7e	~
17	ETB	31	1	4b	K	65	e	7f	D

Table 1 Hexadecimal to Character Conversion Chart (continued)

Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char
18	CAN	32	2	4c	L	66	f		
19	EM	33	3	4d	M	67	g		

Table 2 Conversion of nullcisco-0006.53b7.8e71-fa3/0 To A Client Identifier

00	c	i	s	c	o	-	0	0	0	6	.	5	3	b	7	.	8	e	7	1	-	f	a	3	/	0
00	63	69	73	63	6f	2d	30	30	30	36	2e	35	33	62	37	2e	38	65	37	31	2d	46	61	33	2f	30

R4

Use the **show interface** *interface-type interface-number* command to display the information and statistics for Ethernet 0 on R4.

```
R4> show interface ethernet 0
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 00e0.1eb8.eb0e (bia 00e0.1eb8.eb0e)
```

The MAC address for Ethernet 0 on R4 is 00e0.1eb8.eb0e. The format of the client identifier for this interface is nullcisco-00e0.1eb8.eb0e-et0.

Note The short interface name for Ethernet interfaces is et.

Using the values for converting characters to their hexadecimal equivalents in Table 1, the client identifier for Ethernet 0 on R4 is shown in the last row of Table 3.

Table 3 Conversion of null.cisco-00e0.1eb8.eb0e-et0 To A Client Identifier for R4

00	c	i	s	c	o	-	0	0	e	0	.	1	e	b	8	.	e	b	0	e	-	e	t	0
00	63	69	73	63	6f	2d	30	30	65	30	2e	31	65	62	38	2e	65	62	30	65	2d	45	74	30

R3

Use the **show interface** *interface-type interface-number* command to display the information and statistics for Ethernet 0 on R3.

```
R3> show interface ethernet 0
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 00e0.1eb8.eb73 (bia 00e0.1eb8.eb73)
```

The MAC address for Ethernet 0 on R3 is 00e0.1eb8.eb73. The format of the client identifier for this interface is: nullcisco-00e0.1eb8.eb73-et0.

Using the values for converting characters to their hexadecimal equivalents in Table 1, the client identifier for Ethernet 0 on R3 is shown in the last row of Table 4.

Table 4 Conversion of null.cisco-00e0.1eb8.eb73-et0 To A Client Identifier for R3

00	c	i	s	c	o	-	0	0	e	0	.	1	e	b	8	.	e	b	7	3	-	e	t	0
00	63	69	73	63	6f	2d	30	30	65	30	2e	31	65	62	38	2e	65	62	37	33	2d	45	74	30

R2

Use the **show interface** *interface-type interface-number* command to display the information and statistics for Ethernet 0 on R2.

```
R2> show interface ethernet 0
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 00e0.1eb8.eb09 (bia 00e0.1eb8.eb09)
```

The MAC address for Ethernet 0 on R2 is 00e0.1eb8.eb09. The format of the client identifier for this interface is nullcisco-00e0.1eb8.eb09-et0.

Using the values for converting characters to their hexadecimal equivalents in Table 1, the client identifier for Ethernet 0 on R2 is shown in the last row of Table 5

Table 5 Conversion of null.cisco-00e0.1eb8.eb09-et0 To A Client Identifier for R2

00	c	i	s	c	o	-	0	0	e	0	.	1	e	b	8	.	e	b	0	9	-	e	t	0
00	63	69	73	63	6f	2d	30	30	65	30	2e	31	65	62	38	2e	65	62	30	39	2d	45	74	30

You have now determined the values for the client identifiers on each router. The final step is to add a period after each group of four characters working from the left to the right as shown below:

- R4–0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6230.652d.4574.30
- R3–0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6237.332d.4574.30
- R2–0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6230.392d.4574.30

What to Do Next

Save the values in a text file and proceed to the "Creating a Private DHCP Pool for Each of The Routers" section on page 38.

Determining the Value for the DHCP Client Identifier Automatically

If you determined the value for the client identifiers manually, you do not need to perform this task. Proceed to the "Creating a Private DHCP Pool for Each of The Routers" section on page 38.

This task will create a DHCP server on R1 that will provide only one IP address. This IP address will used by each new router in sequence while you determine the value of the router's client identifier. By limiting the IP address scope to a single IP address you avoid any possible confusion about which router you are working on. If somebody powers up another router that attempts to start the AutoInstall process, it will not be able to obtain an IP address.

Tip Do not place the network-confg or router configuration files (r4-confg, r3-confg, or r2-confg) in the root directory of the TFTP server yet. You do not want any of the routers to load these files until you have ensured that each router will obtain the correct IP address from the DHCP server so that the router will load the correct configuration file.

This task is broken down into sub-tasks to make it easier to follow (all sub-tasks are required):

- Configuring IP on the Interfaces on R1, page 36
- Configuring a DHCP Pool on R1, page 36
- Excluding All But One of the IP Addresses from the DHCP Pool on R1, page 36
- Verifying The Configuration on R1, page 36
- Enabling debug ip dhcp server events on R1, page 37
- Identifying the Value for the Client Identifier on Each of the Routers, page 37
- Removing the DHCP Pool on R1 for Network 172.16.28.0/24, page 38
- Removing the DHCP Pool on R1 for Network 172.16.28.0/24, page 38
- Removing the Excluded Address Range From R1, page 38

Configuring IP on the Interfaces on R1

Configure IP addresses on the Ethernet interfaces. Configure the **ip helper-address** *ip-address* command on Ethernet 0/1.

```
!
interface Ethernet0/0
 ip address 172.16.29.99 255.255.255.0
!
interface Ethernet0/1
 ip address 172.16.28.99 255.255.255.0
 ip helper-address 172.16.29.252
!
```

Configuring a DHCP Pool on R1

Configure these commands to setup the temporary DHCP server on R1.

Note This should be the only DHCP server in operation on R1. This should be the only DHCP server that is accessible by the routers that you will be using AutoInstall to setup.

```
!
ip dhcp pool get-client-id
   network 172.16.28.0 255.255.255.0
!
```

Excluding All But One of the IP Addresses from the DHCP Pool on R1

You need to ensure that there is only one IP address available from the DHCP server at any time. Configure the following command to exclude every IP address except 172.16.28.1 from the DHCP pool.

```
!
ip dhcp excluded-address 172.16.28.2 172.16.28.255
!
```

Verifying The Configuration on R1

Verify that the configuration file for R1 has a DHCP server pool configured to provide a single IP address (172.16.28.1) to a DHCP client.

Verify that the configuration file has the IP addresses for the Ethernet interfaces and the **ip helper-address** *ip-address* command.

```
!
ip dhcp excluded-address 172.16.28.2 172.16.28.255
!
ip dhcp pool get-client-id
   network 172.16.28.0 255.255.255.0
!
interface Ethernet0/0
 ip address 172.16.29.99 255.255.255.0
!
interface Ethernet0/1
 ip address 172.16.28.99 255.255.255.0
 ip helper-address 172.16.29.252
!
```

Enabling debug ip dhcp server events on R1

You use the display output from the **debug ip dhcp server events** command on the terminal connected to R1 to identify the value of the client identifier for each router.

Enable the **debug ip dhcp server events** command on R1.

```
R1# debug ip dhcp server events
```

Identifying the Value for the Client Identifier on Each of the Routers

This step is repeated for each of the routers. You should only have one of the routers powered-on at any time. When you have identified the value of the client identifier field for the router, you will turn the router off and proceed to the next router.

R4

Connect R4 to the Ethernet network and power it on. The following message will be displayed on the terminal connected to R1 when R4 is assigned the IP address 172.16.28.1.

```
DHCPD: assigned IP address 172.16.28.1 to client
0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6230.652d.4574.30.
```

Copy the client identifier 0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6230.652d.4574.30 to a text file and save it. Keep the text file open for the next two routers.

Turn off R4

Release the IP address binding for R4 from the DHCP pool on R1 using the **clear ip dhcp binding *** command on R1.

```
R1# clear ip dhcp binding *
R1#
01:16:11: DHCPD: returned 172.16.28.1 to address pool get-client-id.
```

R3

Connect R3 to the Ethernet network and power it on. The following message will be displayed on the terminal connected to R1 when R3 is assigned the IP address 172.16.28.1.

```
DHCPD: assigned IP address 172.16.28.1 to client
0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6237.332d.4574.30.
```

Copy the client identifier 0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6237.332d.4574.30 to the text file and save it. Keep the text file open for the final router.

Turn off R3.

Release the IP address binding for R3 from the DHCP pool on R1 using the **clear ip dhcp binding *** command on R1.

```
R1# clear ip dhcp binding *
R1#
01:16:11: DHCPD: returned 172.16.28.1 to address pool get-client-id.
```

R2

Connect R2 to the Ethernet network and power it on. The following message will be displayed on the terminal connected to R1 when R2 is assigned the IP address 172.16.28.1.

```
DHCPD: assigned IP address 172.16.28.1 to client
0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6230.392d.4574.30.
```

Copy the client identifier 0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6230.392d.4574.30 to the text file and save it.

Turn off R2

Release the IP address binding for R2 from the DHCP pool on R1 using the **clear ip dhcp binding *** command on R1.

```
R1# clear ip dhcp binding *
R1#
01:16:11: DHCPD: returned 172.16.28.1 to address pool get-client-id.
```

Client Identifiers for R4, R3, and R2

You have determined the values for the client identifiers on each router.

- R4–0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6230.652d.4574.30
- R3–0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6237.332d.4574.30
- R2–0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6230.392d.4574.30

Removing the DHCP Pool on R1 for Network 172.16.28.0/24

The temporary DHCP pool on the router is no longer required, and must be removed.

```
R1(config)# no ip dhcp pool get-client-id
```

Removing the Excluded Address Range From R1

The command for excluding all of the IP addresses except 172.16.28.1 from the DHCP pool on the router is no longer required, and must be removed.

R1(config)# **no ip dhcp excluded-address 172.16.28.2 172.16.28.255**

Creating a Private DHCP Pool for Each of The Routers

You need to create the private DHCP address pools for each router to ensure that each router is assigned the IP address that maps to its host name in the network-conf file.

```
!
ip dhcp pool r4
   host 172.16.28.100 255.255.255.0
   client-identifier 0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6230.652d.4574.30

!
ip dhcp pool r3
```

```
   host 172.16.28.101 255.255.255.0
   client-identifier 0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6237.332d.4574.30

!
ip dhcp pool r2
   host 172.16.28.102 255.255.255.0
   client-identifier 0063.6973.636f.2d30.3065.302e.3165.6238.2e65.6230.392d.4574.30
```

Creating Configuration Files for Each Router

Create the configuration files for each router and place them in the root directory of the TFTP server.

Tip You must include the commands for configuring passwords for remote Telnet access and access to privileged EXEC mode if you are going to access the routers remotely to save their configuration files to NVRAM.

r2-confg

```
!
hostname R2
!
enable secret 7gD2A0
!
interface Ethernet0
 ip address 172.16.28.102 255.255.255.0
!
interface Serial0
 ip address 192.168.100.1 255.255.255.252
 no shutdown
!
interface Serial1
 ip address 192.168.100.5 255.255.255.252
 no shutdown
!
no ip http server
ip classless
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 Ethernet0
!
line vty 0 4
 password 5Rf1k9
 login
!
end
```

r3-confg

```
!
hostname R3
!
enable secret 7gD2A0
!
interface Ethernet0
 ip address 172.16.28.101 255.255.255.0
!
interface Serial0
 ip address 192.168.100.9 255.255.255.252
 no shutdown
!
```

```
interface Serial1
 ip address 192.168.100.13 255.255.255.252
 no shutdown
!
no ip http server
ip classless
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 Ethernet0
!
line vty 0 4
 password 5Rf1k9
 login
!
end
```

r4-confg

```
!
hostname R3
!
enable secret 7gD2A0
!
interface Ethernet0
 ip address 172.16.28.101 255.255.255.0
!
interface Serial0
 ip address 192.168.100.9 255.255.255.252
 no shutdown
!
interface Serial1
 ip address 192.168.100.13 255.255.255.252
 no shutdown
!
no ip http server
ip classless
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 Ethernet0
!
line vty 0 4
 password 5Rf1k9
 login
!
end
```

Creating the network-confg file

Create the network-confg file with the **ip host** *hostname ip-address* commands that map the IP addresses that you will be assigning with the DHCP server to the hostname.

```
ip host r4 172.16.28.100
ip host r3 172.16.28.101
ip host r2 172.16.28.102
```

Setting Up the Routers with AutoInstall

You are now ready to set up the three routers (R4, R3, and R2) using AutoInstall.

Connect a terminal to the routers if you want to monitor the progress of AutoInstall. Use Hyperterminal or a similar terminal emulation program on your PC, with the following terminal emulation settings, to connect to the device:

- 9600 baud
- 8 data bits, no parity, 1 stop bit
- No flow control

You should have the following files in the root directory of the TFTP server.

- network-confg
- r4-confg
- r3-confg
- r2-confg

The TFTP server must be running.

Power on each router.

Timesaver You can set up all three routers concurrently.

R4

The following is an excerpt of the messages that are displayed on R4's console terminal during the AutoInstall process:

```
Loading network-confg from 172.16.29.252 (via Ethernet0): !
[OK - 76 bytes]
Configuration mapped ip address 172.16.28.100 to r4
Loading r4-confg from 172.16.29.252 (via Ethernet0): !
[OK - 687 bytes]
```

R3

The following is an excerpt of the messages that are displayed on R3's console terminal during the AutoInstall process:

```
Loading network-confg from 172.16.29.252 (via Ethernet0): !
[OK - 76 bytes]
Configuration mapped ip address 172.16.28.101 to r3
Loading r3-confg from 172.16.29.252 (via Ethernet0): !
[OK - 687 bytes]
```

R2

The following is an excerpt of the messages that are displayed on R2's console terminal during the AutoInstall process:

```
Loading network-confg from 172.16.29.252 (via Ethernet0): !
[OK - 76 bytes]
Configuration mapped ip address 172.16.28.102 to r2
Loading r2-confg from 172.16.29.252 (via Ethernet0): !
[OK - 687 bytes]
```

TFTP Server Log

The TFTP server log should contain messages similar to the following text.

```
Sent network-confg to (172.16.28.100), 76 bytes
Sent r4-confg to (172.16.28.100),687 bytes
Sent network-confg to (172.16.28.101), 76 bytes
Sent r3-confg to (172.16.28.101),687 bytes
Sent network-confg to (172.16.28.102), 76 bytes
Sent r2-confg to (172.16.28.102),687 bytes
```

Saving the Configuration Files on The Routers

You must save the running configurations on each router to the startup configuration to ensure that the routers retain their configurations if they are ever power cycled.

R4

```
R1# telnet 172.16.28.100
Trying 172.16.28.100 ... Open


User Access Verification

Password:
R4> enable
Password:

R4# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R4# exit

[Connection to 172.16.28.100 closed by foreign host]
R1#
```

R3

```
R1# telnet 172.16.28.101
Trying 172.16.28.101 ... Open


User Access Verification

Password:
R3> enable
Password:

R3# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R3# exit

[Connection to 172.16.28.101 closed by foreign host]
R1#
```

R2

```
R1# telnet 172.16.28.102
Trying 172.16.28.102 ... Open


User Access Verification
```

```
Password:
R2> enable
Password:

R2# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R2# exit

[Connection to 172.16.28.102 closed by foreign host]
R1#
```

Removing the Private DHCP Address Pools from R1

The final step in the AutoInstall process is to remove the private DHCP address pools from R1.

```
R1(config)# no ip dhcp pool r4
R1(config)# no ip dhcp pool r3
R1(config)# no ip dhcp pool r2
```

This is the final task, and step for Using AutoInstall to Setup Devices Connected to LANs.

Using AutoInstall to Set Up Devices Connected to WANs: Example

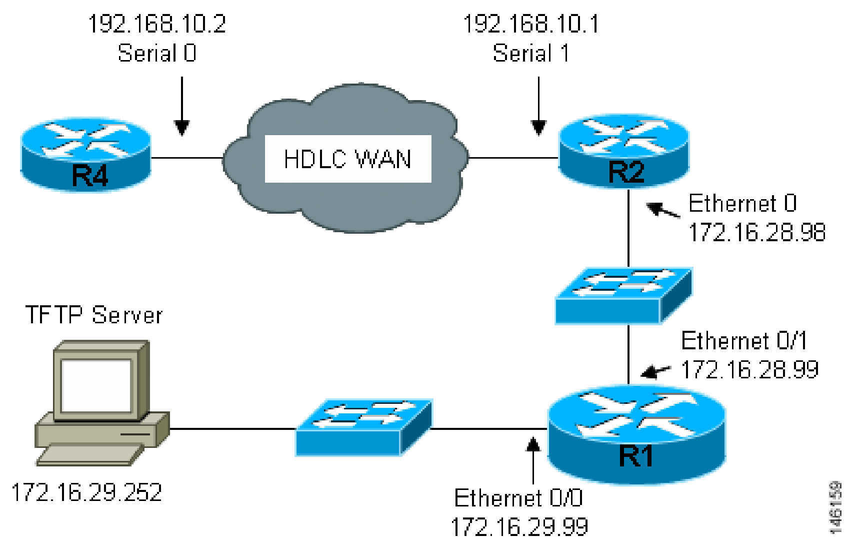
This section contains the following examples:

- HDLC WAN Connections, page 43
- Frame-Relay WAN Connections, page 46

HDLC WAN Connections

This section uses the network in Figure 10. The section shows how to use AutoInstall to setup R4. R2 will use SLARP to provide R4 the IP address (192.168.20.2) required for AutoInstall.

Figure 10 Network Topology Using AutoInstall to Configure Routers Connected to HDLC WANs



The process for using AutoInstall to set up router R2 requires the following tasks:

- Creating the Configuration for R4, page 44
- Creating the network-confg File, page 45
- Configuring R1 and R2, page 45
- Setting Up R4 using AutoInstall, page 46
- Save the Configuration File on R4, page 46

Creating the Configuration for R4

Create the configuration file for R4 and save it on the TFTP server as r4-confg:

```
!
hostname R4
!
enable secret 7gD2A0
!
interface Ethernet0
 ip address 10.89.45.1 255.255.255.0
 no shutdown
!
interface Serial0
 ip address 192.168.10.2 255.255.255.0
 no fair-queue
!
router rip
 version 2
 network 168.192.0.0
 no auto-summary
!
ip http server
ip classless
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 Serial0
!
line vty 0 4
password 6T2daX9
```

```
!
end
```

Creating the network-confg File

Create the network configuration file for R4 and save it on the TFTP server as network-confg:

```
ip host r4 192.168.10.2
```

Configuring R1 and R2

Configure R1 and R2 using the following configurations:

R1

```
!
hostname R1
!
enable secret 7gD2A0
!
interface Ethernet0/0
 ip address 172.16.29.99 255.255.255.0
!
interface Ethernet0/1
 ip address 172.16.28.99 255.255.255.0
 ip helper-address 172.16.29.252
!
router rip
 version 2
 network 172.16.0.0
 no auto-summary
!
ip classless
ip http server
!
line vty 0 4
 password 67F2SaB
!
end
```

R2

```
!
hostname R2
!
enable secret 7gD2A0
!
interface Ethernet0
 ip address 172.16.28.98 255.255.255.0
!
interface Serial1
 ip address 192.168.10.1 255.255.255.0
 clockrate 64000
!
router rip
 version 2
 network 172.16.0.0
 network 192.168.10.0
 no auto-summary
!
ip http server
ip classless
```

```
!
line vty 0 4
 password u58Hg1
!
end
```

Setting Up R4 using AutoInstall

The network is now ready to use AutoInstall to setup R4. perform the following steps to setup R4.

Connect R4 to the HDLC WAN network.

Power R4 on.

The AutoInstall process should be complete in approximately 5 minutes.

TFTP Server Log

The TFTP server log should contain messages similar to the following text:

```
Sent network-confg to (192.168.10.2), 76 bytes
Sent r4-confg to (192.168.10.2),687 bytes
```

Save the Configuration File on R4

You must save the running configurations on R4 to the startup configuration to ensure that R4 retains its configuration if it is ever power cycled.

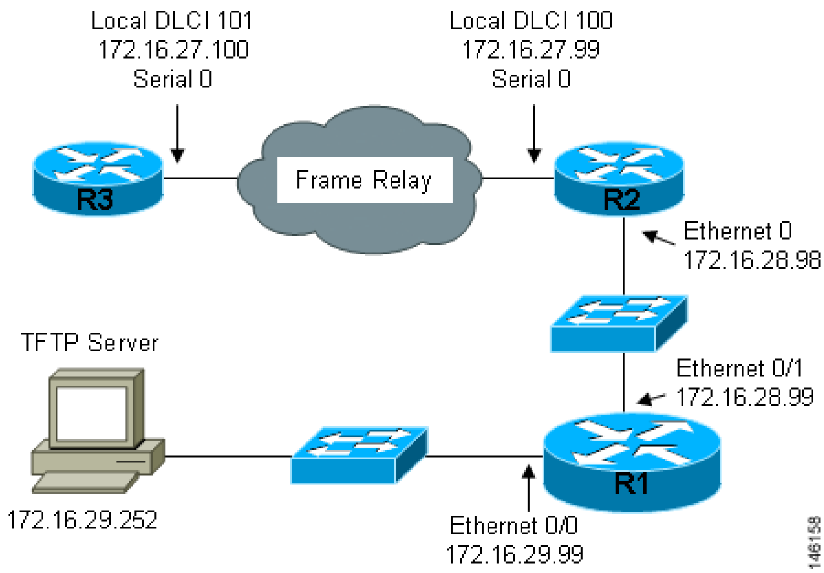
```
R1# telnet 192.168.10.2
Trying 192.168.10.2 ... Open

User Access Verification

Password:
R4> enable
Password:

R4# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R4# exit

[Connection to 192.168.10.2 closed by foreign host]
R1#
```

Frame-Relay WAN Connections

This section uses the network in Figure 11. The section shows how to use AutoInstall to setup R4. R2 will use BOOTP to provide R4 the IP address (172.16.27.100) required for AutoInstall.

R2 uses 172.16.27.100 as the IP address to provide to R3 using BOOTP because this is the IP address in the **frame-relay map ip 172.16.27.100 100 broadcast** command on serial 0 that points to serial 0 on R3.

Figure 11 Network Topology for Using AutoInstall to Configure Routers Connected to Frame Relay WANs



The process for using AutoInstall to set up router R3 requires the following tasks:

- Creating the Configuration for R3
- Creating the network-confg File
- Configuring R1 and R2
- Setting Up R3 using AutoInstall
- Saving the Configuration File on R3

Creating the Configuration for R3

Create the configuration file for R4 and save it on the TFTP server as r3-confg:

```
!
hostname R3
!
enable secret 8Hg5Zc20
!
interface Ethernet0
 no ip address
 shutdown
!
interface Serial0
 ip address 172.16.27.100 255.255.255.0
 encapsulation frame-relay IETF
 no fair-queue
 frame-relay map ip 172.16.27.99 101 broadcast
 frame-relay interface-dlci 101
!
interface Serial1
 no ip address
 shutdown
!
router rip
```

```
 version 2
 network 172.16.0.0
 no auto-summary
!
line vty 0 4
 password 67Td3a
 login
!
end
```

Creating the network-confg File

Create the network configuration file for R3 and save in on the TFTP server as network-confg:

```
ip host r3 172.16.27.100
```

Configuring R1 and R2

Configure R1 and R2 using the following configurations:

R1

```
!
hostname R1
!
enable secret 86vC7Z
!
interface Ethernet0/0
 ip address 172.16.29.99 255.255.255.0
!
interface Ethernet0/1
 ip address 172.16.28.99 255.255.255.0
!
router rip
 version 2
 network 172.16.0.0
 no auto-summary
!
line vty 0 4
 password 6Gu8z0s
!
!
end
```

R2

```
!
hostname R2
!
enable secret 67Hfc5z2
!
interface Ethernet0
 ip address 172.16.28.98 255.255.255.0
 ip helper-address 172.16.29.252
!
interface Serial0
 ip address 172.16.27.99 255.255.255.0
 ip helper-address 172.16.29.252
 encapsulation frame-relay IETF
 no fair-queue
 frame-relay map ip 172.16.27.100 100 broadcast
```

```
 frame-relay interface-dlci 100
!
interface Serial1
 no ip address
!
router rip
 version 2
 network 172.16.0.0
 no auto-summary
!
line vty 0 4
 password 9Jb6Z3g
!
end
```

Setting Up R3 using AutoInstall

The network is now ready to use AutoInstall to set up R3. perform the following steps to setup R4.

Connect R3 to the Frame Relay network.

Power R3 on.

The AutoInstall process should be complete in approximately 5 minutes.

TFTP Server Log

The TFTP server log should contain messages similar to the following text:

```
Sent network-confg to (172.16.27.100), 76 bytes
Sent r3-confg to (172.16.27.100),687 bytes
```

Saving the Configuration File on R3

You must save the running configurations on R3 to the startup configuration to ensure that R3 retains its configuration if it is ever power cycled.

```
R1# telnet 172.16.27.100
Trying 172.16.27.100 ... Open

User Access Verification

Password:
R3> enable
Password:

R3# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R4# exit

[Connection to 192.168.10.2 closed by foreign host]
R1#
```

Additional References

The following sections provide references related to Using AutoInstall to Remotely Configure Cisco Networking Devices.

Related Documents

Related Topic	Document Title
Frame Relay-to-ATM Service Interworking (FRF.8)	Frame Relay-to-ATM Service Interworking (FRF.8) http://www.cisco.com/en/US/products/sw/iosswrel/ps1834/products_feature_guide09186a00800800cb.html Frame Relay-ATM Interworking Supported Standards http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fwan_c/wcfapdx/wcfappa.htm Configuring Frame Relay-ATM Interworking http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fwan_c/wcffratm.htm#15605
Overview of Cisco IOS Setup Mode and AutoInstall for configuring Cisco networking devices	Overview: Basic Configuration of a Cisco Networking Device
Using Setup Mode to Configure a Cisco Networking Device	Using Setup Mode to Configure a Cisco Networking Device

Standards

Standard	Title
FRF. 8.2	"*Frame Relay/ATM PVC Service Interworking Implementation Agreement*" (PDF file) http://www.mae.net/docs/FRF.8.2.pdf

MIBs

MIB	MIBs Link
IF-MIB	The IFNAME object in the IF-MIB can be used to identify the values for the short interface names used in the DHCP Client Identifier for Cisco IOS devices when they are configured as DHCP clients. To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Using AutoInstall to Remotely Configure a Cisco Networking Device

Table 6 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or 12.0(3)S or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 6 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 6 Feature Information for Using AutoInstall to Remotely Set Up a Cisco Networking Device

Feature Name	Releases	Feature Configuration Information
AutoInstall over Frame Relay-ATM Interworking Connections	12.2(4)T	The AutoInstall over Frame Relay-ATM Interworking Connections feature extends the functionality of the existing Cisco IOS AutoInstall feature. While AutoInstall over Frame Relay encapsulated serial interfaces has long been supported, this feature provides the same functionality when the central (existing) router has an ATM interface instead of a Frame Relay interface. The following sections provide information about this feature: • Intermediate Frame Relay-ATM Switching Device (Optional) • Using AutoInstall with Frame Relay to ATM Service Internetworking No new or modified commands are introduced with this feature. All commands used with this feature are documented in the *Cisco IOS Configuration Fundamentals Command Reference.*
AutoInstall Using DHCP for LAN Interfaces	12.1(5)T 12.2(33)SRC Cisco IOS XE Release 2.1	The AutoInstall Using DHCP for LAN Interfaces feature enhances the benefits of AutoInstall by replacing the use of the Bootstrap Protocol (BOOTP) with the use of the Dynamic Host Configuration Protocol (DHCP) for Cisco IOS AutoInstall over LAN interfaces (specifically Ethernet, Token Ring, and FDDI interfaces). In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following section provides information about this feature: • AutoInstall Using DHCP for LAN Interfaces

Configuring Operating Characteristics for Terminals

Configuring Operating Characteristics for Terminals

This chapter describes how to configure operating characteristics for terminals. For a complete description of the terminal operation commands in this chapter, refer to the "Terminal Operating Characteristics Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Terminal Operating Characteristics Configuration Task List

To configure operating characteristics for terminals, perform any of the tasks described in the following sections. All tasks in this chapter are optional.

- Displaying Information About the Current Terminal Session
- Setting Local Terminal Parameters
- Saving Local Settings Between Sessions
- Ending a Session
- Changing Terminal Session Parameters
- Displaying Debug Messages on the Console and Terminals
- Recording the Serial Device Location
- Changing the Retry Interval for a Terminal Port Queue
- Configuring LPD Protocol Support on a Printer

Note For additional information about configuring terminal services, see the Release 12.2 *Cisco IOS Terminal Services Configuration Guide* and the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Displaying Information About the Current Terminal Session

To display terminal line information, use the following commands in user or privileged EXEC mode, as needed:

Command	Purpose
`Router> show whoami text`	Displays information about the terminal line being used for the current session, including host name, line number, line speed, and location. If text is included as an argument in the command, that text is displayed as part of the additional data about the line.
`Router> where`	Lists all open sessions associated with the current terminal line. An asterisk (*) in the output indicates the current terminal session.

The following example shows sample output of the **show whoami** command:

```
Router> show whoami

Comm Server "Router", Line 0 at 0bps.  Location "Second floor, West"

--More--
Router>
```

To prevent the information from disappearing from the screen, the **show whoami** command always displays a --More-- prompt before returning to the CLI prompt. Press the Spacebar to return to the prompt.

Setting Local Terminal Parameters

The **terminal** EXEC mode commands enable or disable features for the current session only. You can use these commands to temporarily change terminal line settings without changing the stored configuration file.

To display a list of the commands for setting terminal parameters for the current session, use the following command in EXEC mode:

Command	Purpose
`Router# terminal ?`	Lists the commands for setting terminal parameters.

The following example shows sample output for the **terminal ?** command. Commands available on your routing device will vary depending on the software image and hardware you are using.

```
Router> terminal ?
  autohangup              Automatically hangup when last connection closes
  data-character-bits     Size of characters being handled
  databits                Set number of data bits per character
  dispatch-character      Define the dispatch character
  dispatch-timeout        Set the dispatch timer
  download                Put line into 'download' mode
  editing                 Enable command line editing
  escape-character        Change the current line's escape character
  exec-character-bits     Size of characters to the command exec
  flowcontrol             Set the flow control
```

```
full-help                Provide help to unprivileged user
help                     Description of the interactive help system
history                  Enable and control the command history function
hold-character           Define the hold character
ip                       IP options
keymap-type              Specify a keymap entry to use
lat                      DEC Local Area Transport (LAT) protocol-specific
                         configuration
length                   Set number of lines on a screen
no                       Negate a command or set its defaults
notify                   Inform users of output from concurrent sessions
padding                  Set padding for a specified output character
parity                   Set terminal parity
rxspeed                  Set the receive speed
special-character-bits   Size of the escape (and other special) characters
speed                    Set the transmit and receive speeds
start-character          Define the start character
stop-character           Define the stop character
stopbits                 Set async line stop bits
telnet                   Telnet protocol-specific configuration
telnet-transparent       Send a CR as a CR followed by a NULL instead of a CR
                         followed by a LF
terminal-type            Set the terminal type
transport                Define transport protocols for line
txspeed                  Set the transmit speeds
width                    Set width of the display terminal
```

Throughout this chapter, many terminal settings can be configured for all terminal sessions or for just the current terminal session. Settings for all terminal sessions are configured in line configuration mode and can be saved. Settings for the current session are specified using EXEC mode commands that generally begin with the word **terminal**.

Saving Local Settings Between Sessions

You can configure the Cisco IOS software to save local parameters (set with **terminal** EXEC mode commands) between sessions. Saving these local settings ensures that the parameters the user sets will remain in effect between terminal sessions. This function is useful for servers in private offices. To save local settings between sessions, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)#` **`private`**	Saves local settings between sessions.

If the **private** line configuration command is not used, user-set terminal parameters are cleared when the session ends with either the **exit** EXEC mode command or when the interval set with the **exec-timeout** line configuration command has passed.

Ending a Session

To end a session, use the following command in EXEC mode:

Command	Purpose
`Router> quit`	Ends the current session.

Refer to the "Managing Connections, Menus, and System Banners" chapter for more information on ending sessions and closing connections.

Changing Terminal Session Parameters

This section explains how to change terminal and line settings both for a particular line and locally. The local settings are set with the **terminal** EXEC mode commands. They temporarily override the settings made by the system administrator and remain in effect only until you exit the system. In line configuration mode, you can set terminal operation characteristics that will be in operation for that line until the next time you change the line parameters.

The following sections describe the tasks used to make the more common changes to the terminal and line settings:

- Defining the Escape Character and Other Key Sequences
- Specifying Telnet Operation Characteristics
- Configuring Data Transparency for File Transfers
- Specifying an International Character Display

The following sections describe the tasks used to make the less common changes to the terminal and line settings:

- Setting Character Padding
- Specifying the Terminal and Keyboard Type
- Changing the Terminal Screen Length and Width
- Enabling Pending Output Notifications
- Creating Character and Packet Dispatch Sequences
- Changing Flow Control for the Current Session
- For more information about setting flow control or to set flow control on a line for more than the current session, refer to the "Configuring Modem Support and Asynchronous Devices" chapter in the Dial Solutions Configuration Guide. For information about X.25 flow control, see the "Configuring X.25 and LAPB" chapter in the "Cisco IOS Wide-Area Networking Configuration Guide".Enabling Session Locking
- Configuring Automatic Baud Rate Detection
- Setting a Line as Insecure
- Configuring Communication Parameters for Terminal Ports

Defining the Escape Character and Other Key Sequences

You can define or modify the default keys used to execute functions for system escape, terminal activation, disconnect, and terminal pause. Generally, the keys used are actually combinations of keys, such as pressing the Control (Ctrl) key and another key (or keys) at the same time (such as Ctrl-^).

Sequences of keys, such as pressing the Control key and another key, then pressing yet another key, are also sometimes used (for example Ctrl-^, x). However, in each case these keys are referred to as characters, because each key or combination of keys is represented by a single ASCII character. For a complete list of available ASCII characters and their decimal and keyboard equivalents, see the "ASCII Character Set" appendix of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Globally Defining Escape Character and Other Key Sequences

To define or change the default key sequences involved with terminal session activation, disconnection, escape, or pausing, use the following commands in line configuration mode, as needed:

Command	Purpose
`Router(config-line)#` **`escape-character`** {*`ascii-number`* | *`ascii-character`* | **`break`** | **`default`** | **`none`**}	Changes the system escape character. We recommend the use of the ASCII characters represented by the decimal numbers 1 through 30. The escape character can be a single character (such as **'**), a key combination (such as Ctrl-X), or a sequence of keys (such as Ctrl-^, X). The default escape character (key combination) is Ctrl-Shift-6 (Ctrl-^), or Ctrl-Shift-6, X (Ctrl-^, X).
`Router(config-line)#` **`activation-character`** *`ascii-number`*	Defines a session activation character. Entering this character at a vacant terminal begins a terminal session. The default activation character is the Return key.
`Router(config-line)#` **`disconnect-character`** *`ascii-number`*	Defines the session disconnect character. Entering this character at a terminal ends the session with the router. There is no default disconnect character.
`Router(config-line)#` **`hold-character`** *`ascii-number`*	Defines the hold character that causes output to the screen to pause. After this character has been set, a user can enter the character at any time to pause output to the terminal screen. To resume output, the user can press any key. To use the hold character in normal communications, precede it with the escape character. There is no default hold character.

For most of the commands described, you can reinstate the default value by using the **no** form. However, to return the escape character to its default, you should use the **escape-character default** line-configuration command.

Note If you are using the autoselect function (enabled using the **autoselect** line configuration command), the activation character should not be changed from the default value of Return. If you change this default, the autoselect feature may not function.

Defining Escape and Pause Characters for the Current Session

For the current terminal session, you can modify key sequences to execute functions for system escape and terminal pause. To modify these sequences, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router> terminal escape-character ascii-number`	Changes the system escape sequence for the current session. The escape sequence indicates that the codes that follow have special meaning. The default key combination is Ctrl-Shift-6 (Ctrl-^).
`Router> terminal hold-character ascii-number`	Defines the hold sequence or character that causes output to the terminal screen to pause for this session. There is no default sequence. To continue the output, type any character after the hold character. To use the hold character in normal communications, precede it with the escape character. You cannot suspend output on the console terminal.

The **terminal escape-character** EXEC command is useful, for example, if you have the default escape character defined for a different purpose in your keyboard file. Entering the escape character followed by the X key returns the router to EXEC mode when the router is connected to another device.

Specifying Telnet Operation Characteristics

To set Telnet operation characteristics for access servers, perform the tasks described in the following sections:

- Generating a Hardware Break Signal for a Reverse Telnet Connection
- Setting the Line to Refuse Full-Duplex, Remote Echo Connections
- Allowing Transmission Speed Negotiation
- Synchronizing the Break Signal
- Changing the End-of-Line Character

Note The commands in this section apply only to access servers.

Generating a Hardware Break Signal for a Reverse Telnet Connection

To cause the access server to generate a hardware Break signal on the EIA/TIA-232 line that is associated with a reverse Telnet connection for the current line and session, use the following command in EXEC mode:

Command	Purpose
`Router> terminal telnet break-on-ip`	Generates a hardware Break signal on the EIA/TIA-232 line that is associated with a reverse Telnet connection for the current line and session.

The hardware Break signal occurs when a Telnet Interrupt-Process command is received on that connection. This command can be used to control the translation of Telnet IP commands into X.25 Break indications.

This command is also a useful workaround in the following situations:

- Several user Telnet programs send an Interrupt-Process command, but cannot send a Telnet Break signal.
- Some Telnet programs implement a Break signal that sends an Interrupt-Process command.

Some EIA/TIA-232 hardware devices use a hardware Break signal for various purposes. A hardware Break signal is generated when a Telnet Break command is received.

Setting the Line to Refuse Full-Duplex, Remote Echo Connections

You can set the line to allow the Cisco IOS software to refuse full-duplex, remote echo connection requests from the other end. This refusal suppresses negotiation of the Telnet Remote Echo and Suppress Go Ahead options. To set the current line to refuse to negotiate full-duplex for the current session or remote echo options on incoming connections, use the following command in EXEC mode:

Command	Purpose
`Router>` **`terminal telnet refuse-negotiations`**	Sets the current line to refuse to negotiate full-duplex for the current session.

Allowing Transmission Speed Negotiation

To allow the Cisco IOS software to negotiate transmission speed for the current line and session, use the following command in EXEC mode:

Command	Purpose
`Router>` **`terminal telnet speed`** *`default-speed maximum-speed`*	Allows the Cisco IOS software to negotiate transmission speed for the current line and session.

You can match line speeds on remote systems in reverse Telnet, on host machines that connect to the network through an access server, or on a group of console lines hooked up to an access server when disparate line speeds are in use at the local and remote ends of the connection. Line speed negotiation adheres to the Remote Flow Control option, defined in RFC 1080.

Synchronizing the Break Signal

You can set lines on the access server to cause a reverse Telnet line to send a Telnet Synchronize signal when it receives a Telnet Break signal. The TCP Synchronize signal clears the data path, but interprets incoming commands. To cause the Cisco IOS software to send a Telnet Synchronize signal when it receives a Telnet Break signal on the current line and session, use the following command in EXEC mode:

Command	Purpose
`Router>` **`terminal telnet sync-on-break`**	Causes the Cisco IOS software to send a Telnet Synchronize signal when it receives a Telnet Break signal on the current line and session.

Changing the End-of-Line Character

The end of each line typed at the terminal is ended with a CR+LF (Carriage Return plus Line Feed) signal. The CR+LF signal is sent when a user presses Enter or Return. To cause the current terminal line to send a CR signal as a CR followed by a NULL instead of a CR followed by a line feed (LF), use the following command in EXEC mode:

Command	Purpose
`Router>` **`terminal telnet transparent`**	Causes the current terminal line to send a CR signal as a CR followed by a NULL instead of a CR followed by an LF.

This command ensures interoperability with different interpretations of end-of-line handling in the Telnet protocol specification.

Configuring Data Transparency for File Transfers

Data transparency enables the Cisco IOS software to pass data on a terminal connection without the data being interpreted as a control character.

During terminal operations, some characters are reserved for special functions. For example, the key combination Ctrl-Shift-6, X (^^x) suspends a session. When transferring files over a terminal connection (using the Xmodem or Kermit protocols, for example), you must suspend the recognition of these special characters to allow a file transfer. This process is called *data transparency.*

You can set a line to act as a transparent pipe so that programs such as Kermit, Xmodem, and CrossTalk can download a file across a terminal line. To temporarily configure a line to act as a transparent pipe for file transfers, use the following command in EXEC mode:

Command	Purpose
`Router>` **`terminal download`**	Configures the terminal line to act as a transparent pipe for file transfers.

The terminal download command is equivalent to using all the following commands:

- **terminal telnet transparent**
- **terminal no escape-character**
- **terminal no hold-character**
- **terminal no padding 0**
- **terminal no padding 128**
- **terminal parity none**
- **terminal databits 8**

Specifying an International Character Display

The classic U.S. ASCII character set is limited to 7 bits (128 characters), which adequately represents most displays in the U.S. Most defaults on the modem router work best on a 7-bit path. However, international character sets and special symbol display can require an 8-bit wide path and other handling.

You can use a 7-bit character set (such as ASCII), or you can enable a full 8-bit international character set (such as ISO 8859). This allows special graphical and international characters for use in banners and prompts, and adds special characters such as software flow control. Character settings can be configured globally, per line, or locally at the user level. Use the following criteria for determining which configuration mode to use when you set this international character display:

- If a large number of connected terminals support nondefault ASCII bit settings, use the global configuration commands.
- If only a few of the connected terminals support nondefault ASCII bit settings, use line configuration commands or the EXEC local terminal setting commands.

Note Setting the EXEC character width to an 8-bit character set can cause failures. If a user on a terminal that is sending parity enters the **help** command, an "unrecognized command" message appears because the system is reading all eight bits, although the eighth bit is not needed for **help**.

If you are using the **autoselect** function, the activation character should be set to the default Return, and the EXEC character bit should be set to 7. If you change these defaults, the application does not recognize the activation request.

Specifying the Character Display for All Lines

To specify a character set for all lines (globally), use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# **default-value exec-character-bits** {**7** | **8**}`	Specifies the character set used in command characters.
`Router(config)# **default-value special-character-bits** {**7** | **8**}`	Specifies the character set used in special characters such as software flow control, hold, escape, and disconnect characters.

Specifying the Character Display for a Line

To specify a character set based on hardware, software, or on a per-line basis, use any of the following commands in line configuration mode:

Command	Purpose
Router(config-line)# **databits** {**5** | **6** | **7** | **8**}	Sets the number of data bits per character that are generated and interpreted by hardware.
Router(config-line)# **data-character-bits** {**7** | **8**}	Sets the number of data bits per character that are generated and interpreted by software.
Router(config-line)# **exec-character-bits** {**7** | **8**}	Specifies the character set used in EXEC and configuration command characters on a per-line basis.
Router(config-line)# **special-character-bits** {**7** | **8**}	Specifies the character set used in special characters (such as software flow control, hold, escape, and disconnect characters) on a per-line basis.

Specifying the Character Display for the Current Session

To specify a character set based on hardware, software, or on a per-line basis for the current terminal session, use the following commands in EXEC mode:

Command	Purpose
`Router>` **`terminal databits`** `{`**`5`** `|` **`6`** `|` **`7`** `|` **`8`**`}`	Sets the number of data bits per character that are generated and interpreted by hardware for the current session.
`Router>` **`terminal data-character-bits`** `{`**`7`** `|` **`8`**`}`	Sets the number of data bits per character that are generated and interpreted by software for the current session.
`Router>` **`terminal exec-character-bits`** `{`**`7`** `|` **`8`**`}`	Specifies the character set used in EXEC and configuration command characters on a per-line basis for the current session.
`Router>` **`terminal special-character-bits`** `{`**`7`** `|` **`8`**`}`	Specifies the character set used in special characters (such as software flow control, hold, escape, and disconnect characters) on a per-line basis for the current session.

Setting Character Padding

Character padding adds a number of null bytes to the end of a line and can be used to make that line an expected length for conformity. You can change the character padding on a specific output character.

Setting Character Padding for a Line

To set character padding for a line, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)#` **`padding`** *`ascii-number count`*	Sets padding on a specific output character for the specified line.

Changing Character Padding for the Current Session

To change character padding on a specific output character for the current session, use the following command in EXEC mode:

Command	Purpose
`Router>` **`terminal padding`** *`ascii-number count`*	Sets padding on a specific output character for the specified line for the current session.

Specifying the Terminal and Keyboard Type

You can specify the type of terminal connected to a line. This feature has two benefits: It provides a record of the type of terminal attached to a line, and it can be used in Telnet terminal negotiations to inform the remote host of the terminal type for display management.

Specifying the Terminal Type for a Line

To specify the terminal type for a line, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# terminal-type {terminal-type}`	Specifies the terminal type. Any string is accepted for the *terminal-type* argument.

This feature is used by TN3270 terminals to identify the keymap and ttycap passed by the Telnet protocol to the end host.

Specifying the Terminal and Keyboard Type for the Current Session

To specify the type of terminal connected to the current line for the current session, use the following command in EXEC mode:

Command	Purpose
`Router> terminal terminal-type terminal-type`	Specifies the terminal type for the current session.

Indicate the terminal type if it is different from the default of VT100. This default is used by TN3270 terminals for display management and by Telnet and rlogin to inform the remote host of the terminal type.

To specify the current keyboard type for a session, use the following command in EXEC mode:

Command	Purpose
`Router> terminal keymap-type keymap-name`	Specifies the keyboard type for the current session.

You must specify the keyboard type when you use a keyboard other than the default of VT100. The system administrator can define other keyboard types (using the **terminal-type** line configuration command) and provide these names to terminal users.

Changing the Terminal Screen Length and Width

By default, the Cisco IOS software provides a screen display of 24 lines by 80 characters. You can change these values if they do not meet the requirements of your terminal. The screen values you set are passed during rsh and rlogin sessions.

The screen values set can be learned by some host systems that use this type of information in terminal negotiation. To disable pausing between screens of output, set the screen length to 0.

The screen length specified can be learned by remote hosts. For example, the rlogin protocol uses the screen length to set terminal parameters on a remote UNIX host. The width specified also can be learned by remote hosts.

Setting the Terminal Screen Length and Width for a Line

To set the terminal screen length and width for all sessions on a line, use either of the following commands in line configuration mode, as needed:

Command	Purpose
`Router(config-line)#` **`length`** *`screen-length`*	Sets the screen length.
`Router(config-line)#` **`width`** *`characters`*	Sets the screen width.

Setting the Terminal Screen Length and Width for the Current Session

To set the number of lines or character columns on the current terminal screen for the current session, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router>` **`terminal length`** *`screen-length`*	Sets the screen length for the current session.
`Router>` **`terminal width`** *`characters`*	Sets the screen width for the current session.

Enabling Pending Output Notifications

You can enable the system to inform users when output is pending on a connection other than the active connection. This feature is for situations in which users are likely to have multiple, concurrent telnet connections through the system. For example, the user might want to know when another connection receives mail or a message.

Enabling Pending Output Notifications for a Line

To enable pending output notifications for a line, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)#` **`notify`**	Enables a line to notify users of pending output on another connection.

Setting Pending Output Notification for the Current Session

To set pending output notification for the current session, use the following command in EXEC mode:

Command	Purpose
`Router>` **`terminal notify`**	Sets up a line to notify a user of pending output for the current session.

Creating Character and Packet Dispatch Sequences

The Cisco IOS software supports dispatch sequences and TCP state machines that send data packets only when they receive a defined character or sequence of characters. You can configure dispatch characters that allow packets to be buffered, then sent upon receipt of a character. You can configure a state machine that allows packets to be buffered, then sent upon receipt of a sequence of characters. This feature enables packet transmission when the user presses a function key, which is typically defined as a sequence of characters, such as Esc I C.

TCP state machines can control TCP processes with a set of predefined character sequences. The current state of the device determines what happens next, given an expected character sequence. The state-machine commands configure the server to search for and recognize a particular sequence of characters, then cycle through a set of states. The user defines these states—up to eight states can be defined. (Think of each state as a task that the server performs based on the assigned configuration commands and the type of character sequences received.)

The Cisco IOS software supports user-specified state machines for determining whether data from an asynchronous port should be sent to the network. This functionality extends the concept of the dispatch character and allows the equivalent of multicharacter dispatch strings.

Up to eight states can be configured for the state machine. Data packets are buffered until the appropriate character or sequence triggers the transmission. Delay and timer metrics allow for more efficient use of system resources. Characters defined in the TCP state machine take precedence over those defined for a dispatch character.

Setting Character and Packet Dispatch Sequences for a Line

To configure your system, use the following commands in line configuration mode:

Command	Purpose
Router(config-line)# **state-machine** *name state firstchar lastchar* [*nextstate* | **transmit**]	Specifies the transition criteria for the states in a TCP state machine.
Router(config-line)# **dispatch-machine** *name*	Specifies the state machine for TCP packet dispatch.
Router(config-line)# **dispatch-character** *ascii-number* [*ascii-number2 . . . ascii-number*]	Defines a character that triggers packet transmission.
Router(config-line)# **dispatch-timeout** *milliseconds*	Sets the dispatch timer.
Router(config-line)# **buffer-length** *length*	Specifies the maximum length of the data stream to be forwarded.

Changing the Packet Dispatch Character for the Current Session

To change the packet dispatch character for the current session, use the following command in EXEC mode:

Command	Purpose
Router> **terminal dispatch-character** *ascii-number1* [*ascii-number2 . . . ascii-number*]	Defines a character that triggers packet transmission for the current session.

Changing Flow Control for the Current Session

To change flow control between the router and attached device for this session, use the following commands in EXEC mode, as needed:

Command	Purpose
Router> **terminal flowcontrol** {**none** | **software** [**in** | **out**] | **hardware**}	Sets the terminal flow control for this session.
Router> **terminal start-character** *ascii-number*[1]	Sets the flow control start character in the current session.
Router> **terminal stop-character** *ascii-number*[1]	Sets the flow control stop character in the current session.

1. This command is seldom used. Typically, you only need to use the **terminal flowcontrol** command.

For more information about setting flow control or to set flow control on a line for more than the current session, refer to the "Configuring Modem Support and Asynchronous Devices" chapter in the *Dial Solutions Configuration Guide*. For information about X.25 flow control, see the "Configuring X.25 and LAPB" chapter in the "Cisco IOS Wide-Area Networking Configuration Guide".

Enabling Session Locking

The **lock** EXEC command temporarily locks access to a session, denying access to other users. Session locking must be enabled on the line for the **lock** command to work. To allow session locking by users on a specific line or group of lines, use the following command in line configuration mode:

Command	Purpose
Router(config-line)# **lockable**	Enables a temporary terminal-locking mechanism.

Configuring Automatic Baud Rate Detection

You can configure a line to automatically detect the baud rate being used. To set up automatic baud rate detection, use the following command in line configuration mode:

Command	Purpose
Router(config-line)# **autobaud**	Configures a line to automatically detect the baud rate.

Note Do not use the **autobaud** command with the **autoselect** command.

To start communications using automatic baud detection, use multiple Returns at the terminal. A 600-, 1800-, or 19200-baud line requires three Returns to detect the baud rate. A line at any other baud rate requires only two Returns. If you use extra Returns after the baud rate is detected, the EXEC facility simply displays another system prompt.

Setting a Line as Insecure

You can set up a terminal line to appear as an insecure dialup line. The information is used by the local-area transport (LAT) software, which reports such dialup connections to remote systems.

To set a line as insecure, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)#` **`insecure`**	Sets the line as a dialup line.

In early releases of Cisco IOS software, any line that used modem control was reported as dialup connection through the LAT protocol; this command allows more direct control of your line.

Configuring Communication Parameters for Terminal Ports

You can change the following parameters as necessary to meet the requirements of the terminal or host to which you are attached. To do so, use the following commands in EXEC mode, as needed:

Command	Purpose
Router> **terminal** {**speed** | **txspeed** | **rxspeed**} *bps*	Sets the line speed for the current session. Choose from line speed, transmit speed, or receive speed.
Router> **terminal databits** {**5** | **6** | **7** | **8**}	Sets the data bits for the current session.
Router> **terminal stopbits** {**1** | **1.5** | **2**}	Sets the stop bits for the current session.
Router> **terminal parity** {**none** | **even** | **odd** | **space** | **mark**}	Sets the parity bit for the current session.

Displaying Debug Messages on the Console and Terminals

To display **debug** command output and system error messages in EXEC mode on the current terminal, use the following command in privileged EXEC mode:

Command	Purpose
Router# **terminal monitor**	Displays **debug** command output and system error messages in EXEC mode on the current terminal.

Remember that all terminal parameter-setting commands are set locally and do not remain in effect after a session is ended. You must use this command at the privileged-level EXEC prompt at each session to display the debugging messages.

Recording the Serial Device Location

You can record the location of a serial device. The text provided for the location appears in the output of the EXEC monitoring commands. To record the device location, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)#` **`location`** *`text`*	Records the location of a serial device.

Changing the Retry Interval for a Terminal Port Queue

If you attempt to connect to a remote device such as a printer that is busy, the connection attempt is placed in a terminal port queue. If the retry interval is set too high, and several routers or other devices are connected to the remote device, your connection attempt can have long delays. To change the retry interval for a terminal port queue, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`terminal-queue entry-retry-interval`** *`interval`*	Changes the retry interval for a terminal port queue.

Configuring LPD Protocol Support on a Printer

The Cisco IOS software supports a subset of the Berkeley UNIX Line Printer Daemon (LPD) protocol used to send print jobs between UNIX systems. This subset of the LPD protocol permits the following:

- Improved status information
- Cancellation of print jobs
- Confirmation of printing and automatic retry for common print failures
- Use of standard UNIX software

The Cisco implementation of LPD permits you to configure a printer to allow several types of data to be sent as print jobs (for example, PostScript or raw text).

To configure a printer for the LPD protocol, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`printer`** *`printername`* `{`**`line`** *`number`* `|` **`rotary`** *`number`*`} [`**`newline-convert`**`]`	Configures a printer and specifies a tty line (or lines) for the device.

If you use the **printer** command, you also must modify the /etc/printcap file on the UNIX system to include the definition of the remote printer on the router. Use the optional **newline-convert** keyword on UNIX systems that do not handle single character line terminators to convert a new line to a character Return, line-feed sequence.

The following example includes the configuration of the printer named saturn on the host memphis:

```
commlpt|Printer on cisco AccessServer:\
```

```
:rm=memphis:rp+saturn:\
:sd+/usr/spool/lpd/comm1pt:\
:lf=?var/log/lpd/commlpt:
```

The content of the actual file may differ, depending on the configuration of your UNIX system.

To print, users use the standard UNIX lpr command.

Support for the LPD protocol allows you to display a list of currently defined printers and current usage statistics for each printer. To do so, use the following command in EXEC mode:

Command	Purpose
`Router>` **`show printer`**	Lists currently defined printers and their current usage statistics.

To provide access to LPD features, your system administrator must configure a printer and assign a TTY line (or lines) to the printer. The administrator must also modify the /etc/printcap file on your UNIX system to include the definition of the remote printer in the Cisco IOS software.

Managing Connections, Menus, and System Banners

Managing Connections, Menus, and System Banners

This chapter describes how to manage connections to other hosts, set banner messages for router users, and create menus of specific user tasks.

The tasks in this document use commands that initially became available in Cisco IOS Release 12.2. Additional supplemental documentation may be available for later and derivative releases. To locate detailed documentation of commands that appear in this chapter, use *Cisco IOS Release 12.4 Master Indexes*.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature. For more information, see the "About Cisco IOS Software Documentation" chapter.

Managing Connections, Menus, and System Banners Task List

To manage connections, configure messages and banners, and create user menus, perform any of the tasks described in the following sections, as needed. All tasks in this chapter are optional.

- Managing Connections, page 2
- Configuring Terminal Messages, page 7
- Enabling Terminal Banners, page 8
- Creating Menus, page 12

Examples for these sections can be found at the end of the chapter in the "Connection Management, System Banner, and User Menu Configuration Examples" section.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Managing Connections

To configure connection-management activities that apply to all supported connection protocols, perform the tasks described in the following sections. All tasks are optional.

- Displaying Current Terminal Settings, page 2
- Escaping Terminal Sessions and Switching to Other Connections, page 3
- Assigning a Logical Name to a Connection, page 3
- Changing a Login Username
- Locking Access to a Terminal, page 5
- Sending Messages to Other Terminals, page 5
- Clearing TCP Connections, page 6
- Exiting a Session Started from a Router, page 6
- Logging Out of a Router, page 6
- Disconnecting a Line, page 7

Displaying Current Terminal Settings

To display the current settings for the terminal line connection, use the following command in privileged or user EXEC mode:

Command	Purpose
`Router# show terminal`	Displays current settings for the terminal.

The following example shows sample output:

```
AccessServer1> show terminal

Line 2, Location: "", Type: "VT220"
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 9600/9600
Status: PSI Enabled, Ready, Active, No Exit Banner
Capabilities: none
Modem state: Ready
Group codes:    0
Special Chars: Escape  Hold  Stop  Start  Disconnect  Activation
                ^^x    none   -      -       none
Timeouts:      Idle EXEC    Idle Session   Modem Answer  Session   Dispatch
               00:10:00       never                        none     not set
                            Idle Session Disconnect Warning
                              never
                            Login-sequence User Response
                             00:00:30
                            Autoselect Initial Wait
                              not set
Modem type is unknown.
Session limit is not set.
Time since activation: 00:01:07
Editing is enabled.
History is enabled, history size is 10.
DNS resolution in show commands is enabled
```

```
Full user help is disabled
Allowed transports are lat pad v120 mop telnet rlogin nasi.  Preferred is lat.
No output characters are padded
No special data dispatching characters
```

Escaping Terminal Sessions and Switching to Other Connections

After you have started a connection, you can escape out of the current terminal session by using the escape key sequence (Ctrl-Shift-6 then X by default). You can type the command character as you hold down the Ctrl key or with the Ctrl key released; you can type either uppercase or lowercase letters.

Note In screen output examples that show two caret (^^) symbols together, the first caret represents the Control key (Ctrl) and the second caret represents the key sequence Shift-6. The double-caret combination (^^) means hold down the Ctrl key while you press the Shift and the 6 key.

By default, the escape key sequence is Ctrl-Shift-6, X. However, the escape key sequence can be changed using the **escape-character** line configuration command. To determine the current setting for the escape character, use the **show terminal** privileged or user EXEC command.

You can have several concurrent sessions open and switch back and forth between them.

The number of sessions that can be open at one time is defined by the **session-limit V**DPN configuration mode command.

To switch between sessions by escaping one session and resuming a previously opened session, perform the following steps:

Step 1 Escape out of the current session by pressing the escape key sequence (Ctrl-Shift-6 then X [Ctrl^, X] by default) and return to the EXEC prompt.

Step 2 Enter the **where privileged EXEC** command to list the open sessions. All open sessions associated with the current terminal line are displayed.

Step 3 Enter the **resume** privileged EXEC command and the session number to make the connection.

You also can resume the previous session by pressing the Return key.

The Ctrl^, X key combination and the **where** and **resume** privileged EXEC commands are available with all supported connection protocols (for example, Telnet).

Assigning a Logical Name to a Connection

To assign a logical name to a connection, use the following command in user EXEC mode:

Command	Purpose
`Router#` **`name-connection`**	Assigns a logical name to a connection.

The logical name can be useful for keeping track of multiple connections.

You are prompted for the connection number and name to assign. The **where** privileged EXEC command displays a list of the assigned logical connection names.

Changing a Login Username

You can change your login username if you must match outgoing access list requirements or other login prompt requirements. A login server must be running and available to use this command. To change a login username, use the following command in user EXEC mode:

Command	Purpose
`Router> login`	Allows you to log in to the system a second time for the purposes of changing your login name.

When you enter this command, the system prompts you for a username and password. Enter the new username and the original password. If the username does not match, but the password does, the Cisco IOS software updates the session with the new username used by the **login** command attempt. For example, assume that a user logged in as user1 needs to change the login name to user2:

```
Router> login
Username: user2
Password: <letmein>
Router>
```

In this example, the password letmein is the same password used at the initial login. (The angle brackets in the example indicate that the password is not displayed on the screen when entered.) At the second `Router>` prompt, the user is now logged in as user2.

If no username and password prompts appear, the network administrator did not specify that a username and password be required at login time. If both the username and password are entered correctly, the session becomes associated with the specified username.

To access a system with TACACS security, enter your login name or specify a TACACS server by using the *user@tacacs-server* syntax when the "Username:" prompt appears, as shown in the following steps:

	Command	Purpose
Step 1	`Router> login`	Allows you to log in to the system a second time for the purposes of changing your login name.
Step 2	`Username: user@tacacs-server`	Specifies the new username and authenticates the name with the server specified with the *tacacs-server* argument.
Step 3	`Password: <password>`	Specifies the TACACS password for the username specified in Step 2.

Only the specified host (tacacs-server) is accessed for user authentication information.

In the following example, user2 specifies the TACACS host host1 to authenticate the password:

```
Router> login
Username: user2@host1
Translating "HOST1"...domain server (131.108.1.111) [OK]
Password: <letmein2>
```

If you do not specify a host, the router tries each of the TACACS servers in the list until it receives a response. If you specify a host that does not respond, no other TACACS server will be queried. The router either will deny access or it will function, according to the action specified by the **tacacs-server last-resort** global configuration command, if it is configured. If you specified a TACACS server host with the *user@tacacs-server* argument, the TACACS server specified is used for all subsequent authentication or notification queries, with the possible exception of Serial Line Internet Protocol (SLIP) address queries.

For more information on configuring TACACS, refer to the **tacacs-server host** global configuration command in the "TACACS, Extended TACACS, and TACACS+ Commands" chapter of the *Cisco IOS Security Command Reference*.

For an example of changing a login name, see the "Changing a Login Username and Password: Example" section at the end of this chapter.

Locking Access to a Terminal

You can prevent access to your terminal session while keeping your connection open by setting a temporary password. For this temporary locking feature to work, the line must first be configured to allow locking (using the **lockable** line-configuration mode command). To lock access to the terminal, perform the following steps:

Step 1 Issue the **lock** command in user or privileged EXEC mode.

When you issue this command, the system will prompt you for a password.

Step 2 Enter a password, which can be any arbitrary string. The system will prompt you to confirm the password. The screen then is cleared, and the message "Locked" is displayed.

Step 3 To regain access to your session, reenter the password.

The Cisco IOS software honors session timeouts on locked lines. You must clear the line to remove this feature.

The following is an example of the prompts displayed after the **lock** command is entered. Note that the entered password does not appear on screen.

```
Router# lock
Password:
Again:
                                    Locked
Password:
Router#
```

Sending Messages to Other Terminals

You can send messages to one or all terminals. A common reason for doing this is to inform users of an impending shutdown. To send a message to other terminals, use the following command in user EXEC or privileged EXEC mode:

Command	Purpose
`Router#` **`send`** `{`*`line-number`* `|` **`*`**`}`	Sends a message to other terminals. Using the * sends messages to all terminals.

The system prompts for the message, which can be up to 500 characters long. Press Ctrl-Z to end the message. Press Ctrl-C to abort the command.

Clearing TCP Connections

To clear a TCP connection, use the following command in privileged EXEC mode:

Command	Purpose
`Router#` **`clear tcp`** `{`**`line`** *`line-number`* `|` **`local`** *`host-name port`* **`remote`** *`host-name port`* `|` **`tcb`** *`tcb-address`*`}`	Clears a TCP connection.

The **clear tcp** command is particularly useful for clearing non-functioning TCP connections.

The **clear tcp line** *line-number* command terminates the TCP connection on the specified tty line. All TCP sessions initiated from that tty line are also terminated.

The **clear tcp local** *host-name port* **remote** *host-name port* command terminates the specific TCP connection identified by the hostname/port pair of the local and remote router.

Exiting a Session Started from a Router

The protocol used to initiate a session determines how you exit that session.

To exit from SLIP and PPP connections, you must hang up the dial-in connection, usually with a command that your dial-in software supports.

To exit a local area transport (LAT), Telnet, rlogin, TN3270, or X.3 packet assembler/disassembler (PAD) session begun from the router to a remote device, press the escape key sequence (Ctrl-Shift-6 then X [Ctrl^X] by default for some systems, Ctrl-Z by default for other systems) and enter the **disconnect** command at the EXEC prompt. You can also log out of the remote system.

You can use either the **exit** or **logout** command in EXEC mode to terminate an active terminal session.

To exit a Telnet session *to* a router, see the "Logging Out of a Router" section, which follows.

Logging Out of a Router

The method you use to logout from or disconnect from a router depends on where you are located in relation to the router, and the port on the router to which you log in.

If your terminal or computer running a terminal-emulation application is remotely connected to the console port of the router, you disconnect by issuing the command or key sequence used by your terminal-emulation package. For example, if you are on a Macintosh computer running the application TCP/Connect from InterCon Corporation, you would press Ctrl-] at the user or privileged EXEC prompt to disconnect.

If you are on a remote terminal and connect to a vty through a synchronous interface on the router, you can issue one of the following commands in user EXEC or privileged EXEC mode to log out:

- **exit**
- **logout**

Disconnecting a Line

Note Avoid disconnecting a line to end a session. Instead, log out of the host to allow the router to clear the connection. You should disconnect a line only if you cannot log out of an active session (for example, if the line is stuck or frozen).

To disconnect a line, use the following command in EXEC mode:

Command	Purpose
`Router# **disconnect** [*connection*]`	Disconnects a line.

If your terminal or computer running a terminal-emulation application is connected physically to the console port of the router, you can also disconnect from the router by physically disconnecting the cable from the console port of the router.

Configuring Terminal Messages

To configure messages that can be displayed to terminal users that connect to the system, perform any of the tasks found in the following sections. All tasks are optional.

- Enabling an Idle Terminal Message, page 7
- Configuring a "Line in Use" Message, page 8
- Configuring a "Host Failed" Message, page 8

Enabling an Idle Terminal Message

You can configure the system to display a message when a console or terminal is not in use. Also called a *vacant message*, this message is different from the banner message displayed when a user logs in to the system. To enable the idle terminal message, use the following command in line configuration mode:

Command	Purpose
Router(config-line)# **vacant-message** [*d message d*]	Configures the system to display an idle terminal message. The argument *d* indicates any delimiting character.

Tip Commands requiring a delimiting character (the *d* argument) are used throughout this chapter. Any character can be used as the delimiting character, but we recommend the use of the quote sign ("), because this character is unlikely to be needed within the message itself. Other commonly used delimiting characters include the percent sign (%) or the forward slash (/), but because these characters have meanings within certain Cisco IOS commands, they are not recommended. For example, to set the vacant message to `This terminal is idle` you would enter the command **vacant-message " This terminal is idle "**.

Configuring a "Line in Use" Message

To configure the system to display a "line in use" message when an incoming connection is attempted and all rotary group or other lines are in use, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)#` **`refuse-message`** *`d message d`*	Configures the system to display a "line in use" message. The argument *d* indicates any delimiting character.

If you do not define such a message, the user receives a system-generated error message when all lines are in use. You also can use this message to provide the user with further instructions.

Configuring a "Host Failed" Message

To configure the system to display a "host failed" message when a Telnet connection with a specific host fails, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)#` **`busy-message`** *`hostname d message d`*	Configures the system to display a "host failed" message. The argument *d* indicates any delimiting character.

Enabling Terminal Banners

Banners are informational messages that can be displayed to users. To enable terminal banners, perform any of the tasks in the following sections. All tasks are optional.

- Configuring a Message-of-the-Day Banner, page 9
- Configuring a Login Banner, page 10
- Configuring an EXEC Banner, page 10
- Configuring a Banner Sent on Incoming Connections, page 10
- Configuring a SLIP-PPP Banner Message, page 11
- Enabling or Disabling the Display of Banners, page 11

For an example of displaying terminal banner messages, see the "Configuring Banners: Example" section at the end of this chapter.

Using Banner Tokens

Banners can be customized with the use of banner tokens. Tokens are keywords in the form $(*token*) that, when used in a banner message, display the currently configured value of the token argument (for example, the router hostname, domain name, or IP address). Using these tokens, you can design your own banners that will display current Cisco IOS configuration variables. Only Cisco IOS supported tokens may be used. There is no facility for you to define your own tokens.

Table 8 lists the tokens supported by the different **banner** commands.

Table 8 Tokens Allowed by Banner Type

Token	Description	motd banner	login banner	exec banner	incoming banner	slip-ppp banner
$(hostname)	Router Hostname	Yes	Yes	Yes	Yes	Yes
$(domain)	Router Domain Name	Yes	Yes	Yes	Yes	Yes
$(peer-ip)	IP Address of the Peer Machine	No	No	No	No	Yes
$(gate-ip)	IP Address of the Gateway Machine	No	No	No	No	Yes
$(encap)	Encapsulation Type (SLIP or PPP)	No	No	No	No	Yes
$(encap-alt)	Encapsulation Type Displayed as SL/IP instead of SLIP	No	No	No	No	Yes
$(mtu)	Maximum Transmission Unit Size	No	No	No	No	Yes
$(line)	vty or tty (async) Line Number	Yes	Yes	Yes	Yes	No
$(line-desc)	User-specified description of the Line	Yes	Yes	Yes	Yes	No

Configuring a Message-of-the-Day Banner

You can configure a message-of-the-day (MOTD) banner to be displayed on all connected terminals. This banner is displayed at login and is useful for sending messages (such as impending system shutdowns) that affect all network users. To do so, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`banner motd`** *`d message d`*	Configures the system to display a message-of-the-day banner. The argument *d* indicates any delimiting character.

Configuring a Login Banner

You can configure a login banner to be displayed on all connected terminals. This banner is displayed after the MOTD banner appears and before the login prompts.

To configure a login banner, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`banner login`** *`d message d`*	Configures the system to display a banner before the username and password login prompts. The argument *d* indicates any delimiting character.

The login banner cannot be disabled on a per-line basis. To globally disable the login banner, you must delete the login banner with the **no banner login** command.

Configuring an EXEC Banner

You can configure a banner to be displayed whenever an EXEC process is initiated. For example, this banner will be displayed to a user using Telnet to access the system after entering a username and password, but before the user EXEC mode prompt is displayed. To configure an EXEC banner, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`banner exec`** *`d message d`*	Configures the system to display a banner whenever an EXEC process is initiated. The argument *d* indicates any delimiting character.

Configuring a Banner Sent on Incoming Connections

You can configure a banner to be displayed on terminals connected to reverse Telnet lines. This banner is useful for providing instructions to users of these types of connections. Reverse Telnet connections are described in more detail in the "Configuring and Managing External Modems" chapter of the Release 12.4 *Cisco IOS Dial Technologies Configuration Guide*.

To configure a banner that is sent on incoming connections, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`banner incoming`** *`d message d`*	Configures the system to display a banner when there is an incoming connection to a terminal line from a host on the network. The argument *d* indicates any delimiting character.

Configuring a SLIP-PPP Banner Message

Default banner messages have been known to cause connectivity problems in some non-Cisco SLIP and PPP dialup software. You can customize the SLIP-PPP banner message to make Cisco SLIP and PPP compatible with non-Cisco dialup software. To configure a SLIP-PPP banner message, use the following command in global configuration mode:

Command	Purpose
Router(config)# **banner slip-ppp** *d message d*	Configures a SLIP-PPP banner to display a customized message. The argument *d* indicates any delimiting character.

Enabling or Disabling the Display of Banners

You can control display of the MOTD and line-activation (EXEC) banners. By default, these banners are displayed on all lines. To enable or disable the display of such banners, use the following commands in line configuration mode, as needed:

Command	Purpose
Router(config-line)# **no exec-banner**	Suppresses the display of MOTD and EXEC banners.
Router(config-line)# **exec-banner**	Reinstates the display of the EXEC or MOTD banners.
Router(config-line)# **no motd-banner**	Suppresses the display of MOTD banners.
Router(config-line)# **motd-banner**	Reinstates the display of the MOTD banners.

These commands determine whether the router will display the EXEC banner and the MOTD banner when an EXEC session is created. These banners are defined with the **banner motd** and **banner exec** global configuration commands. By default, the MOTD banner and the EXEC banner are enabled on all lines.

Disable the EXEC and MOTD banners using the **no exec-banner** command.

The MOTD banners can also be disabled by the **no motd-banner** line configuration command, which disables MOTD banners on a line. If the **no exec-banner** command is configured on a line, the MOTD banner will be disabled regardless of whether the **motd-banner** command is enabled or disabled. Table 9 summarizes the effects of the combination of the **exec-banner** command and the **motd-banner** command.

Table 9 Banners Displayed by exec-banner and motd-banner Command Combinations

	exec-banner (default)	**no exec-banner**
motd-banner (default)	MOTD banner EXEC banner	None
no motd-banner	EXEC banner	None

For reverse Telnet connections, the EXEC banner is never displayed. Instead, the incoming banner is displayed. The MOTD banner is displayed by default, but it is disabled if either the **no exec-banner** command or **no motd-banner** command is configured. Table 10 summarizes the effects of the combination of the **exec-banner** command and the **motd-banner** command for reverse Telnet connections.

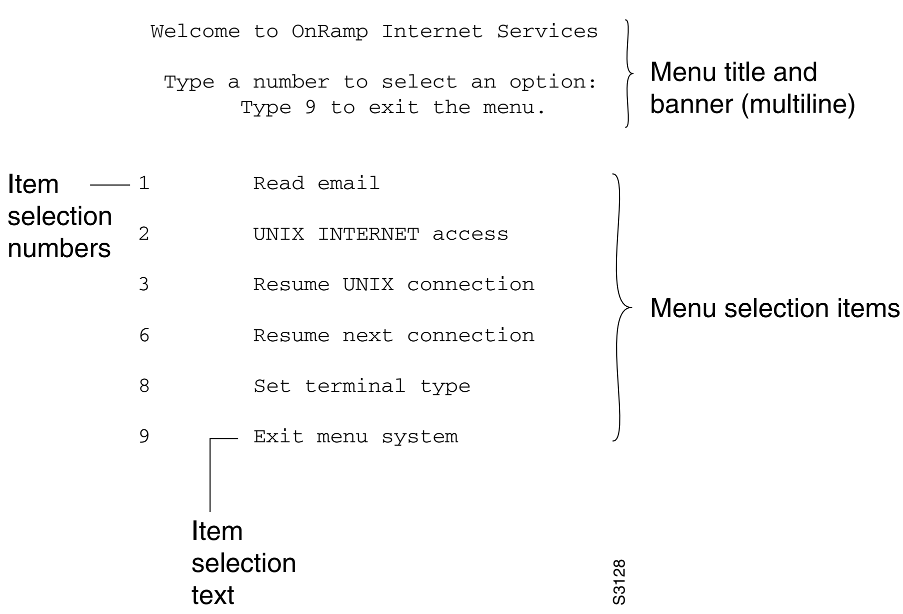
Table 10 ***Banners Displayed Based on exec-banner and motd-banner Command Combinations for Reverse Telnet Sessions to Async Lines***

	exec-banner (default)	**no exec-banner**
motd-banner (default)	MOTD banner Incoming banner	Incoming banner
no motd-banner	Incoming banner	Incoming banner

Creating Menus

A menu is a displayed list of actions from which a user can select without needing to know anything about the underlying command-level details. A menu system (also known as a user menu) effectively controls the functions a user can access. Figure 6 illustrates the parts that make up a typical menu.

Figure 6 ***Typical Menu Example***



Any user that can enter configuration mode can create menus. Remember the following guidelines when you create menus:

- Each menu item represents a single user command.
- The menu system default is a standard "dumb" terminal that displays text only in a 24-line-by-80-column format.

- A menu can have no more than 18 menu items. Menus containing more than 9 menu items are automatically configured as single-spaced menus; menus containing 9 or fewer menu items are automatically configured as double-spaced menus, but can be configured as single-spaced menus using the **menu single-space** global configuration command. (For more information about menu display configuration options, see the section "Specifying Menu Display Configuration Options" later in this chapter.)
- Item keys can be numbers, letters, or strings. If you use strings, you must configure the **menu line-mode** global configuration command.
- When you construct a menu, always specify how a user exits a menu and where the user goes. If you do not provide an exit from a menu—such as with the **menu-exit** command (described in the section "Specifying the Underlying Command for the Menu Item" later in this chapter), the user will be trapped.

The **exec-timeout** line configuration command can be used to close and clean up an idle menu; the **session-timeout** command can be used to clean up a menu with an open connection.

Creating a Menu Task List

To create menus, perform the tasks described in the following sections:

- Specifying the Menu Title, page 13 (Required)
- Specifying the Menu Prompt, page 15 (Optional)
- Specifying the Menu Item Text, page 15 (Required)
- Specifying the Underlying Command for the Menu Item, page 15 (Required)
- Specifying the Default Command for the Menu, page 17 (Required)
- Creating a Submenu, page 17 (Optional)
- Creating Hidden Menu Entries, page 18 (Optional)
- Specifying Menu Display Configuration Options, page 19 (Optional)
- Specifying per-Item Menu Options, page 20 (Optional)
- Invoking the Menu, page 20 (Required)
- Deleting the Menu from the Configuration, page 21 (Optional)

Specifying the Menu Title

You can specify an identifying title for the menu. To specify the menu title, use the following command in global configuration mode:

Command	Purpose
`Router(config)# menu menu-name title d title d`	Specifies the title for the menu. The argument *d* indicates any delimiting character.

The following example specifies the title that is displayed when the OnRamp menu is selected. The following four main elements create the title:

- The **menu title** command

- Delimiter characters that open and close the title text
- Escape characters to clear the screen (optional)
- Title text

The following example shows the command used to create the title for the menu shown in Figure 6:

```
Router(config)# menu OnRamp title %^[[H^[[J

Enter TEXT message.  End with the character '%'.
        Welcome to OnRamp Internet Services

        Type a number to select an option;
              Type 9 to exit the menu.
%
Router(config)#
```

You can position the title of the menu horizontally by preceding the title text with blank characters. You can also add lines of space above and below the title by pressing Enter.

In this example, the title text consists of the following elements:

- One-line title
- Space
- Two-line menu instruction banner

Title text must be enclosed within text delimiter characters—the percent sign character (%) in this example. Title text delimiters are characters that do not ordinarily appear within the text of a title, such as slash (/), double quote ("), or tilde (~). You can use any character that is not likely to be used within the text of the title as delimiter characters. Ctrl-C is reserved for special use and should not be used in the text of the title.

This title text example also includes an escape character sequence to clear the screen before displaying the menu. In this case the string `^[[H^[[J` is an escape string used by many VT100-compatible terminals to clear the screen. To enter it, you must enter Ctrl-V before each escape character (^[).

You can also use the **menu clear-screen** global configuration command to clear the screen before displaying menus and submenus, instead of embedding a terminal-specific string in the menu title. This option uses a terminal-independent mechanism based on termcap entries defined in the router and the terminal type configured for the user terminal. The **menu clear-screen** command allows the same menu to be used on multiple types of terminals instead of terminal-specific strings being embedded within menu titles. If the termcap entry does not contain a clear string, the menu system inserts 24 new lines, causing all existing text to scroll off the top of the terminal screen.

To clear the screen before displaying the menu, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **menu** *menu-name* **clear-screen**	Specifies screen clearing before displaying menus and submenus.

The following example clears the screen before displacing the OnRamp menu or a submenu:

```
Router(config)# menu OnRamp clear-screen
```

Specifying the Menu Prompt

To specify a menu prompt, use the following command in global configuration mode:

Command	Purpose
Router(config)# **menu** *menu-name* **prompt** *d prompt d*	Specifies the prompt for the menu. The argument *d* indicates any delimiting character.

Specifying the Menu Item Text

Each displayed menu entry consists of the selection key (number, letter, or string) and the text describing the action to be performed. You can specify descriptive text for a maximum number of 18 menu items. Because each menu entry represents a single user interface command, you must specify the menu item text one entry at a time. To specify the menu item text, use the following command in global configuration mode:

Command	Purpose
Router(config)# **menu** *menu-name* **text** *menu-item menu-text*	Specifies the text for the menu item.

The following example specifies the text that is displayed for the three entries in the OnRamp menu:

```
Router(config)# menu OnRamp text 1 Read email
Router(config)# menu OnRamp text 2 UNIX Internet Access
Router(config)# menu OnRamp text 9 Exit menu system
```

You can provide access to context-sensitive help by creating a "help server" host and using a menu entry to make a connection to that host.

Menu selection keys need not be contiguous. You can provide consistency across menus by assigning a particular number, letter, or string to a special function—such as Help or Exit—regardless of the number of menu entries in a given menu. For example, menu entry H could be reserved for help across all menus.

When more than nine menu items are defined in a menu, the **menu line-mode** and **menu single-space** global configuration commands are activated automatically. The commands can be configured explicitly for menus of nine items or fewer. For more information on these commands, see the section "Specifying Menu Display Configuration Options" later in this chapter.

Specifying the Underlying Command for the Menu Item

Each displayed menu entry issues a user interface command when the user enters its key. Each menu entry can have only a single command associated with it. To specify the underlying menu item command, use the following command in global configuration mode:

Command	Purpose
Router(config)# **menu** *menu-name* **command** *menu-item command*	Specifies the command to be performed when the menu item is selected.

The following example specifies the commands that are associated with the three entries in the OnRamp menu:

```
Router(config)# menu OnRamp command 1 rlogin mailsys
Router(config)# menu OnRamp command 2 rlogin unix.cisco.com
Router(config)# menu OnRamp command 9 menu-exit
```

The **menu-exit** command is available only from within menus. This command provides a way to return to a higher-level menu or to exit the menu system.

When a menu item allows you to make a connection, the menu item should also contain entries that can be used to resume connections; otherwise, when you try to escape from a connection and return to the menu, there is no way to resume the session. It will sit idle until you log out.

You can build the **resume connection** user EXEC command into a menu entry so that the user can resume a connection, or you can configure the line using the **escape-char none** command to prevent users from escaping their sessions.

To specify connection resumption as part of the menu item command, use the following command in global configuration mode:

Command	Purpose
Router(config)# **menu** *menu-name* **command** *menu-item* **resume** [*connection*] **/connect** [*connect string*]	Specifies that the **resume** command will be performed when the menu item is selected.

Embedding the **resume** command within the **menu** command permits a user to resume the named connection or make another connection using the specified name, if there is no active connection by that name. As an option, you can also supply the connect string needed to connect initially. When you do not supply this connect string, the command uses the specified connection name.

You can use the **resume** command in the following menu entries:

- Embedded in a menu entry
- As a separate, specific menu entry
- As a “rotary” menu entry that steps through several connections

In the following example, the **resume** command is embedded in the **menu** command so that selecting the menu item either starts the specified connection session (if one is not already open) or resumes the session (if one is already open):

```
Router(config)# menu newmenu text 1 Read email
Router(config)# menu newmenu command 1 resume mailsys /connect rlogin mailsys
```

In the following example, the **resume** command is used in a separate menu entry (entry 3) to resume a specific connection:

```
Router(config)# menu newmenu text 3 Resume UNIX Internet Access
Router(config)# menu newmenu command 3 resume unix.cisco.com
```

You use the **resume/next** command to resume the next open connection in the user list of connections. This command allows you to create a single menu entry that advances through all of the user connections. To specify **resume/next** connection resumption as part of the menu item command, use the following command in global configuration mode:

Command	Purpose
Router(config)# **menu** *menu-name* **command** *menu-item* **resume/next**	Specifies **resume/next** connection resumption.

The following example shows a menu entry (entry 6) created to advance through all of the user connections:

```
Router(config)# menu newmenu text 6 Resume next connection
Router(config)# menu newmenu command 6 resume/next
```

Specifying the Default Command for the Menu

When a user presses Enter without specifying an item, the router performs the command for the default item. To specify the default item, use the following command in global configuration mode:

Command	Purpose
Router(config)# **menu** *menu-name* **default** *menu-item*	Specifies the command to be performed when the menu user does not select a menu item.

Creating a Submenu

To create submenus that are opened by selecting a higher-level menu entry, use the **menu** command to invoke a menu in a line menu entry. To specify a submenu item command, use the following commands in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **menu** *menu-name* **text** *menu-item menu-text*	Specifies the menu item that invokes the submenu.
Step 2	Router(config)# **menu** *menu-name* **command** *menu-item* **menu** *menu-name2*	Specifies the command to be used when the menu item is selected.
Step 3	Router(config)# **menu** *menu-name* **title** *delimiter menu-title delimiter*	Specifies the title for the submenu.
Step 4	Router(config)# **menu** *menu-name* **text** *menu-item menu-text*	Specifies the submenu item.
Step 5	Router(config)# **menu** *menu-name* **command** *menu-item command*	Specifies the command to be used when the submenu item is selected. Repeat this command as needed.

The following example specifies that the menu item (entry 8) activates the submenu in the OnRamp menu:

```
Router(config)# menu OnRamp text 8 Set terminal type
```

The following example specifies the command that is performed when the menu item (entry 8) is selected in the OnRamp menu:

```
Router(config)# menu OnRamp command 8 menu Terminals
```

The following example specifies the title for the Terminals submenu:

```
Router(config)# menu Terminals title /
              Supported Terminal Types

          Type a number to select an option;
         Type 9 to return to the previous menu.
```

The following example specifies the submenu items for the Terminals submenu:

```
Router(config)# menu Terminals text 1 DEC VT420 or similar
Router(config)# menu Terminals text 2 Heath H-19
Router(config)# menu Terminals text 3 IBM 3051 or equivalent
Router(config)# menu Terminals text 4 Macintosh with gterm emulator
Router(config)# menu Terminals text 9 Return to previous menu
```

The following example specifies the commands associated with the items in the Terminals submenu:

```
Router(config)# menu Terminals command 1 term terminal-type vt420
Router(config)# menu Terminals command 2 term terminal-type h19
Router(config)# menu Terminals command 3 term terminal-type ibm3051
Router(config)# menu Terminals command 4 term terminal-type gterm
Router(config)# menu Terminals command 9 menu-exit
```

When you select entry 8 on the main menu, the following Terminals submenu appears:

```
      Supported Terminal Types

  Type a number to select an option;
Type 9 to return to the previous menu.

1      DEC VT420 or similar

2      Heath H-19

3      IBM 3051 or equivalent

4      Macintosh with gterm emulator

9      Return to previous menu
```

Note If you nest too many levels of menus, the system displays an error message on the terminal and returns to the previous menu level.

Creating Hidden Menu Entries

A hidden menu entry is a menu item that contains a selection key but no associated text describing the action to be performed. Include this type of menu entry to aid system administrators that provide help to users. The normal procedure is to specify a menu command but omit specifying any text for the item. To create a hidden menu item, use the following command in global configuration mode:

Command	Purpose
Router(config)# **menu** *menu-name* **command** *menu-item command*	Specifies the command to be used when the hidden menu entry is selected.

The following example shows the command associated with the submenu entry in the OnRamp menu:

```
Router(config)# menu OnRamp command 7 show whoami
```

If additional text is appended to the **show whoami** command, that text is displayed as part of the data about the line. For example, the hidden menu entry created by the command

```
Router(config)# menu OnRamp command 7 show whoami Terminals submenu of OnRamp Internet
Access menu
```

will display information similar to the following:

```
Comm Server "cs101", Line 0 at 0 bps. Location "Second floor, West"
Additional data: Terminals submenu of OnRamp Internet Access menu
```

To prevent the information from being lost if the menu display clears the screen, this command always displays a --More-- prompt before returning.

Specifying Menu Display Configuration Options

In addition to the **menu clear-screen** global configuration command (described in the "Specifying the Menu Title" section), the following three **menu** commands define menu functions:

- **menu line-mode**
- **menu single-space**
- **menu status-line**

Configuring the Menu to Operate in Line Mode

In a menu of nine or fewer items, you ordinarily select a menu item by entering the item number or a letter. In line mode, you select a menu entry by entering the item key and pressing Enter. The line mode allows you to backspace over the selection and enter another before pressing Enter to issue the command. This function allows you to change the selection before you invoke the command.

To configure the menu to operate in line mode, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`menu`** *`menu-name`* **`line-mode`**	Configures the menu to use line mode for entering menu items.

The line-mode option is invoked automatically when more than nine menu items are defined, but it can also be configured explicitly for menus of nine items or fewer.

In order to use strings as selection keys, you must enable the **menu line-mode** command.

Displaying Single-Spaced Menus

If there are nine or fewer menu items, the Cisco IOS software ordinarily displays the menu items double-spaced. In a menu of more than nine items, the **single-space** option is activated automatically to fit the menu into a normal 24-line terminal screen. However, the single-space option also can be configured explicitly for menus of nine or fewer items.

To use the **single-space** option to display single-spaced menus, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`menu`** *`menu-name`* **`single-space`**	Configures the specified menu to display single-spaced.

Displaying an Informational Status Line

The **status-line** option displays a line of status information about the current user at the top of the terminal screen before the menu title is displayed. This status line includes the router host name, the user line number, and the current terminal type and keymap type (if any).

To display the **informational status line**, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`menu`** *`menu-name`* **`status-line`**	Configures the specified menu to display a status line.

Specifying per-Item Menu Options

To configure per-item menu options, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)#` **`menu`** *`menu-name`* **`options`** *`menu-item`* **`pause`**	Configures the system to pause after the specified menu item is selected by the user. Enter this command once for each menu item that pauses.
`Router(config)#` **`menu`** *`menu-name`* **`options`** *`menu-item`* **`login`**	Configures the specified menu item to require a login before executing the command. Enter this command once for each menu item that requires a login.

Invoking the Menu

To invoke (access) a menu, use the following command in user EXEC or privileged EXEC mode:

Command	Purpose
`Router#` **`menu`** *`menu-name`*	Invokes a preconfigured user menu.

You can define menus containing privileged EXEC commands, but users must have privileged access when they start up the menu.

To ensure that a menu is automatically invoked on a line, make sure the menu does not have any exit paths that leave users in an interface they cannot operate, then configure that line with the **autocommand menu** *menu-name* line configuration command. (The **autocommand menu** *menu-name* command configures the line to automatically execute the **menu** *menu-name* command when a user initiates a connection over that line.)

Menus also can be invoked on a per-user basis by defining an **autocommand** command for that local username.

In the following example, the OnRamp menu is invoked:

```
Router# menu OnRamp

      Welcome to OnRamp Internet Services

       Type a number to select an option;
```

```
          Type 9 to exit the menu.

1      Read email

2      UNIX Internet access

3      Resume UNIX connection


6      Resume next connection

9      Exit menu system
```

Deleting the Menu from the Configuration

To delete the menu from the configuration, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no menu menu-name`	Deletes the menu by specifying the menu name.

In order to use the menu again, you must reconfigure the entire menu.

The following example deletes the OnRamp menu from the configuration:

```
Router(config)# no menu OnRamp
```

Connection Management, System Banner, and User Menu Configuration Examples

This section provides the following examples:

- Changing a Login Username and Password: Example, page 21
- Sending Messages to Other Terminals: Example, page 22
- Clearing a TCP/IP Connection: Example, page 22
- Configuring Banners: Example, page 23
- Configuring a SLIP-PPP Banner Message, page 11
- Configuring a Menu: Example, page 24

Changing a Login Username and Password: Example

The following example shows how login usernames and passwords can be changed. In this example, a user currently logged in under the username user1 attempts to change that login name to user2. After entering the **login** command, the user enters the new username, but enters an incorrect password. Because the password does not match the original password, the system rejects the attempt to change the username.

```
Router> login
Username: user2
Password:
```

```
% Access denied
Still logged in as "user1"
```

Next, the user attempts the login change again, with the username user2, but enters the correct (original) password. This time the password matches the current login information, the login username is changed to user2, and the user is allowed access to the user login information.

```
Router> login
Username: user2
Password:
Router>
```

Sending Messages to Other Terminals: Example

```
The following example shows the process of sending a message to all terminal connections
on the router:

Router# send *
Enter message, end with CTRL/Z; abort with CTRL/C:
this is a message^Z
Send message? [confirm]
Router#


***
***
*** Message from tty50 to all terminals:
***
this is a message


Router#
```

Clearing a TCP/IP Connection: Example

The following example clears a TCP connection using its tty line number. The **show tcp** EXEC command displays the line number (tty2) that is used in the **clear tcp privileged** EXEC command mode.

```
Router# show tcp

    tty2, virtual tty from host router20.cisco.com
    Connection state is ESTAB, I/O status: 1, unread input bytes: 0
    Local host: 171.69.233.7, Local port: 23
    Foreign host: 171.69.61.75, Foreign port: 1058

    Enqueued packets for retransmit: 0, input: 0, saved: 0

    Event Timers (current time is 0x36144):
    Timer          Starts    Wakeups            Next
    Retrans             4          0             0x0
    TimeWait            0          0             0x0
    AckHold             7          4             0x0
    SendWnd             0          0             0x0
    KeepAlive           0          0             0x0
    GiveUp              0          0             0x0
    PmtuAger            0          0             0x0
```

```
    iss: 4151109680  snduna: 4151109752  sndnxt: 4151109752     sndwnd:  24576
    irs: 1249472001  rcvnxt: 1249472032  rcvwnd:       4258  delrcvwnd:     30

    SRTT: 710 ms, RTTO: 4442 ms, RTV: 1511 ms, KRTT: 0 ms
    minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 300 ms

Router# clear tcp line 2

    [confirm]
     [OK]
```

The following example clears a TCP connection by specifying its local router hostname and port and its remote router hostname and port. The **show tcp brief** privileged EXEC command displays the local (Local Address) and remote (Foreign Address) hostnames and ports to use in the **clear tcp** privileged EXEC command.

```
Router# show tcp brief

    TCB       Local Address           Foreign Address        (state)
    60A34E9C  router1.cisco.com.23      router20.cisco.1055  ESTAB

Router# clear tcp local router1 23 remote router20 1055

    [confirm]
     [OK]
```

The following example clears a TCP connection using its TCB address. The **show tcp brief** EXEC command displays the TCB address to use in the **clear tcp** EXEC command.

```
Router# show tcp brief

    TCB       Local Address           Foreign Address        (state)
    60B75E48  router1.cisco.com.23      router20.cisco.1054  ESTAB

Router# clear tcp tcb 60B75E48

    [confirm]
     [OK]
```

Configuring Banners: Example

The following example shows how to use the **banner** global configuration commands to notify your users that the server will be reloaded with new software. The **no exec-banner** line configuration command is used to disable EXEC banners and message-of-the-day banners on the vty lines.

```
!
line vty 0 4
 no exec-banner
!
banner exec /
 This is Cisco Systems training group router.

 Unauthorized access prohibited.
 /
!
banner incoming /
 You are connected to a Hayes-compatible modem.

 Enter the appropriate AT commands.
 Remember to reset anything you have changed before disconnecting.
 /
```

```
!
banner motd /
 The router will go down at 6pm today for a software upgrade
 /
```

When someone connects to the router, the MOTD banner appears before the login prompt. After the user logs in to the router, the router will display the EXEC banner or incoming banner, depending on the type of connection. For a reverse Telnet login, the router will display the incoming banner. For all other connections, the router will display the EXEC banner.

Configuring a SLIP-PPP Banner with Banner Tokens: Example

The following example configures the SLIP-PPP banner using several tokens and the percent sign (%) as the delimiting character:

```
Router(config)# banner slip-ppp %

Enter TEXT message.  End with the character '%'.
Starting $(encap) connection from $(gate-ip) to $(peer-ip) using a maximum packet size of
$(mtu) bytes... %
```

When a user enters the **slip** command, that user will see the following banner. Notice that the **$(***token***)** syntax is replaced by the corresponding configuration variable.

```
Starting SLIP connection from 192.168.69.96 to 172.16.80.8 using a maximum packet size of
1500 bytes...
```

Configuring a Menu: Example

The following example allows menu users to use Telnet to access one of three different machines. The user also can display the output of the **show user** EXEC command and exit the menu. One hidden menu item (configured as `menu new command here show version`) allows system administrators to display the current software version.

```
menu new title ^C


              Telnet Menu


^C
menu new prompt ^C

Please enter your selection: ^C
menu new text 1 telnet system1
menu new command 1 telnet system1
menu new options 1 pause
menu new text 2 telnet system2
menu new command 2 telnet system2
menu new options 2 pause
menu new text b telnet system3
menu new command b telnet system3
menu new options b pause
menu new text me show user
menu new command me show user
menu new options me pause
menu new command here show version
menu new text Exit Exit
```

```
menu new command Exit menu-exit
menu new clear-screen
menu new status-line
menu new default me
menu new line-mode
!
```

Using the Cisco Web Browser User Interface

Using the Cisco Web Browser User Interface

The Cisco IOS software includes a Web browser user interface (UI) from which you can issue Cisco IOS commands. The Cisco IOS Web browser UI is accessed from the router home page, and can be customized for your business environment. For example, you can view pages in different languages and save them in Flash memory for easy retrieval.This chapter discusses the tasks associated with using and customizing the Cisco Web browser UI.

For a complete description of the Cisco Web browser UI configuration commands in this chapter, refer to the "Cisco IOS Web Browser User Interface Commands" chapter of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

Cisco Web Browser UI Task List

You can issue most Cisco IOS commands using a Web browser by connecting to the home page generated by the Cisco IOS software for your system. Most Cisco routers and access servers automatically generate a password protected home page when the HTTP server is enabled on the device. To access the home page, your computer must be on the same network as the router.

To use the Cisco Web browser UI, your computer must have a World Wide Web browser application. The Cisco Web browser UI works with most web browsers, including Internet Explorer and Netscape Navigator. Your Web browser must be able to read and submit forms.

To use the Cisco Web browser UI, perform the tasks in the following sections:

- Enabling the Cisco Web Browser UI (Required)
- Configuring Access to the Cisco Web Browser UI (Required)
- Accessing and Using the Cisco Web Browser UI (Required)
- Customizing the Cisco Web Browser UI (Optional)

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Enabling the Cisco Web Browser UI

The Web browser UI is automatically enabled on the Cisco 1003, Cisco 1004, or Cisco 1005 router to allow you to use ClickStart to configure your router. For all other Cisco devices, you must enable the Cisco Web browser UI as described here.

To enable the Cisco Web browser UI, you must enable the HTTP server on your router. To enable the HTTP server, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`ip http server`**	Enables the HTTP server (web server) on the system.

Configuring Access to the Cisco Web Browser UI

To control access to the Cisco Web browser UI, you can specify the authentication method for the HTTP server, apply an access list to the HTTP server, and assign a port number for the HTTP server, as described in the following sections.

Specifying the Method for User Authentication

To specify how HTTP server users are authenticated, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`ip http authentication`** `{`**`aaa`** `|` **`enable`** `|` **`local`** `|` **`tacacs`**`}`	Specifies how the HTTP server users are authenticated.

The **ip http authentication** command specifies the authentication method to be used for login when a client connects to the HTTP server. Use of the **ip http authentication aaa** command option is recommended. The **enable**, **local**, and **tacacs** methods should be specified using the **aaa authentication login** command.

If you do not use this command, the default authentication method is used. The default method of authentication for the HTTP server is to use the configured "enable" password. The "enable" password is configured with the **enable password** global configuration command. If the enable password is used as the HTTP server login authentication method, the client connects to the HTTP server with a default privilege level of 15.

Note When the "enable" password is used as the HTTP server login authentication method, any username entered will be ignored; the server will only verify the "enable" password. This may make it easier for an attacker to access the router. Because a username and password pair is more secure than using only a password for authentication, using only "enable" password for authentication is strongly discouraged. Instead, use of the **local** or **tacacs** authentication options, configured as part of a global Authentication, Authorization, and Accounting (AAA) framework, is recommended.
To configure HTTP access as part of a AAA policy, use the **ip http authentication aaa** command option. The "local", "tacacs", or "enable" authentication methods should then be configured using the **aaa authentication login** command.

For information about adding users into the local username database, refer to the *Cisco IOS Security Configuration Guide*.

Example: Configuring the HTTP Server Authentication Method

The following example specifies that the method configured for AAA should be used for authentication for HTTP server users. The AAA login method is configured as the "local" username/password authentication method.

```
Router(config)# ip http authentication aaa
Router(config)# aaa authentication login default local
```

Applying an Access List to the HTTP Server

To control which hosts can access the HTTP server used by the Cisco Web browser UI, you can apply an access list to the HTTP server. To apply an access list to the HTTP server, use the following command in global configuration mode:

Command	Purpose
Router(config)# **ip http access-class** {*access-list-number* | *access-list-name*}	Applies an access list to the HTTP server used by the Cisco IOS ClickStart software or the Cisco Web browser user interface.

Example: Configuring an Access List for HTTP Server Access

In the following example the access list identified as "20" is defined and assigned to the HTTP server:

```
Router(config)# ip access-list standard 20
Router(config-std-nacl)# permit 209.165.202.0 0.0.0.255
Router(config-std-nacl)# permit 209.165.0.0 0.0.255.255
Router(config-std-nacl)# permit 209.0.0.0 0.255.255.255
! (Note: all other access implicitly denied)
Router(config-std-nacl)# exit
Router(config)# ip http access-class 20
```

Changing the HTTP Server Port Number

By default, the HTTP server uses port 80 on the router. To assign the Cisco Web browser UI to a different port, use the following command in global configuration mode:

Command	Purpose
Router(config)# **ip http port** *number*	Assigns a port number to be used by the Cisco Web browser interface.

Accessing and Using the Cisco Web Browser UI

This section describes the tasks used to access the Cisco Web browser UI and issue commands.

Accessing the Router Home Page

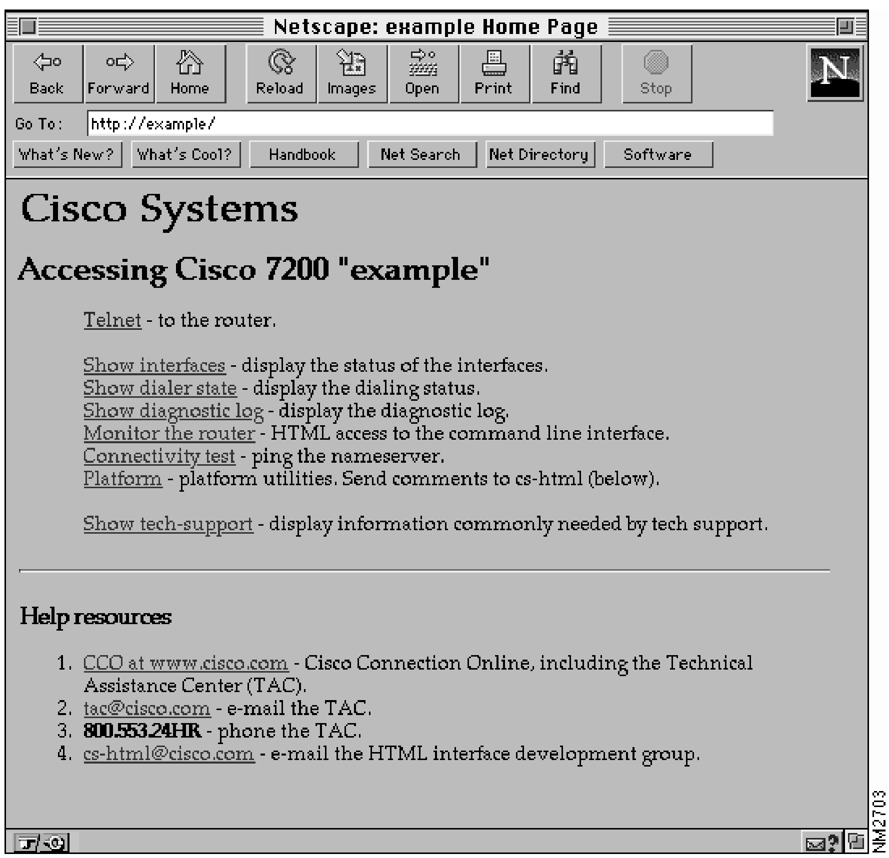
To access a router home page, perform the following steps:

Step 1 Enter **http://***router-name***/** in the URL field of your Web browser and press **Return** . (For example, to access a Cisco router named cacophony, type **http://cacophony/**.) The browser then prompts you for the password.

Step 2 Enter the password. The required password is dependent on the user authentication method configured for the HTTP server (using the **ip http authentication** global configuration command).

After entering the password, the browser will display the router home page. An example of a router home page is shown in shown in Figure 7.

Figure 7 Example of a Home Page for a Cisco 7200 Series Router



The default privilege level when accessing a router home page is privilege level 15 (global access). If privilege levels have been configured on the router and you have been assigned a privilege level other than 15, you must specify the privilege level to access the router home page.

When you specify a privilege level, the Cisco Web Browser UI will display and accept only those commands that have been defined for your user level. (For more information about privilege levels, see the "Configuring Passwords and Privileges" chapter in the Release 12.2 *Cisco IOS Security Configuration Guide*.)

To access a router Web page for a preassigned privilege level other than the default of 15, perform the following steps:

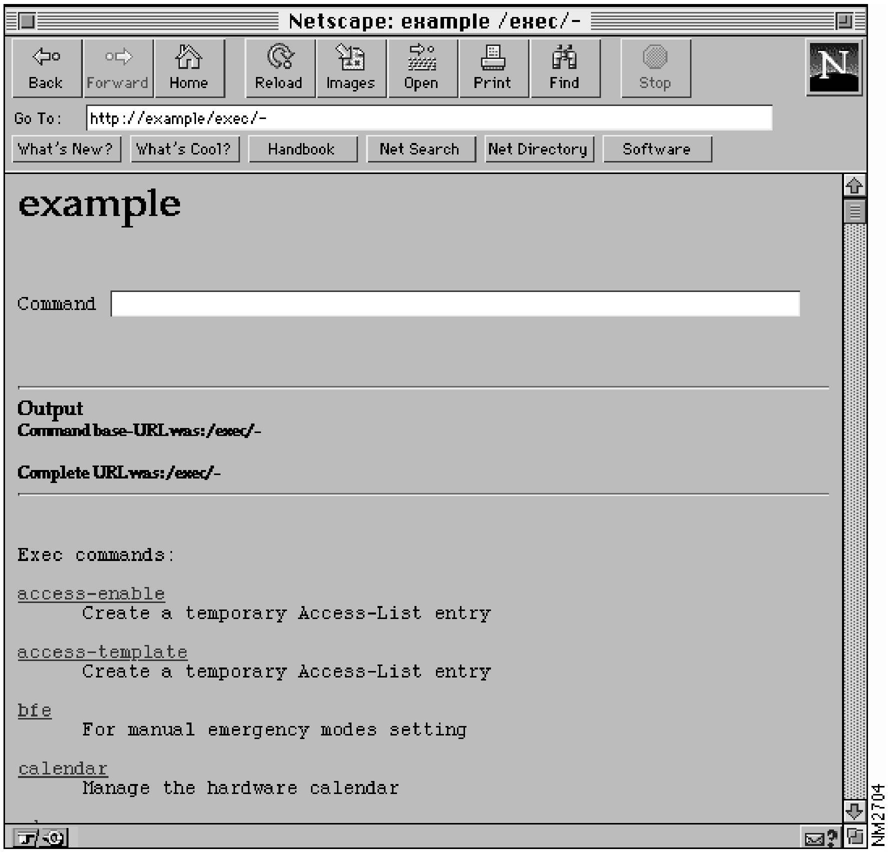
Step 1 Enter **http://***router-name***/level/***level***/exec** in the URL field of your Web browser and press **Return**. For example, to request access to EXEC mode at user privilege level of 12 on a Cisco router named cacophony, type **http://cacophony/level/12/exec**. The browser will then prompt you for your username and password.

Step 2 Enter your username and password and press **Return**. The required password is dependent on the user authentication method configured for the HTTP server. The Web browser will display a Web page specific to your user privilege level.

Issuing Commands Using the Cisco Web Browser UI

From the router home page, click the hypertext link titled **Monitor the Router**. This link takes you to a Web page that has a Command field. An example is shown in Figure 8. You can enter commands in the command field in the same way as you would enter commands using the Cisco IOS command-line interface. The page also displays a list of commands. You can execute these commands by clicking them, as if you were clicking hypertext links.

Figure 8 The Command Field Web Page for a Router Named example



Entering Commands Using Hypertext Links

To enter a command using hypertext links, scroll through the commands listed at the bottom of the screen and click the one you want to execute. If the link is a complete command, it is executed. If the command has more parameters, another list of command hypertext links is displayed. Scroll through this second list and click the one you want to execute.

If the command is a request for information, like a **show** EXEC command, the information is displayed in the Web browser window.

If the command requires a variable, a form in which you can enter the variable is displayed.

Entering Commands Using the Command Field

Entering the command in the command field is just like entering it at a terminal console. Enter the command using the syntax documented in the Cisco IOS command reference. If you are uncertain of the options available for a particular command, type a question mark (**?**).

For example, entering **show ?** in the command field displays the parameters for the **show** EXEC command. The Cisco Web browser UI displays the parameters as hypertext links. To select a parameter, you can either click on one of the links or you can enter the parameter in the command field.

Entering Commands Using the URL Window

You can issue a command using the URL window for the Web browser. To issue a command using the URL window, use the following syntax:

http://*router-name***/**[**level/***level***/**]*command-mode***/***command*

Table 13 lists the URL arguments you must use when requesting a web page.

Table 13 Web Browser URL Argument Descriptions

Argument	Description
router-name	Name of the router being configured.
level/*level*	(Optional) The privilege level you are requesting at which you are requesting access.
mode	The mode the command will be executed in, such as EXEC, configuration, or interface.
command	The command you want to execute. Replace spaces in the command syntax with forward slashes. If you do not specify a command in the URL, your browser will display a web page listing all of the commands available for the specified command mode.

For example, to execute a **show running-configuration** EXEC command on a router named example, you would enter the following in the URL window:

```
http://example/exec/show/running-configuration
```

After issuing this command, the Cisco Web browser UI will display the running configuration for the router.

The difference between entering a command in the Command field and entering a command in the URL window is that in the URL window, forward slashes should be used instead of spaces in the command syntax.

Customizing the Cisco Web Browser UI

You can customize the HTML pages used by the Cisco Web browser UI to display Cisco IOS command output and Cisco IOS platform-specific variables (for example, a router host name or router address). You can display this information using HTML formatted Server Side Includes (SSIs) that you insert into your custom HTML pages. See primarily FEAT-106 (IOS Internationalization) and FEAT-108 (HTTP Security) in PDS. See also Functional Spec ENG-11035 in EDCS. For future plans, see ENG-84169.

Understanding SSIs

SSIs are HTML formatted commands or variables that you insert into HTML pages when you customize Cisco IOS platform configuration pages for a Web browser. These SSI commands and SSI variables display Cisco IOS command output and Cisco IOS platform-specific variables.

Note The majority of the customization features in this section are for the ClickStart EZsetup feature for the Cisco 1000 series, Cisco 1003/1004 series, and Cisco 1005 series routers only.

The Cisco IOS software supports two HTML SSI commands defined for customizing HTML pages: the SSI EXEC command and the SSI ECHO command. The HTML format of the SSI EXEC command is **<!--#exec cmd="*xxx*"-->**, and the HTML format of the SSI ECHO command is **<!--#echo var="*yyy*"-->**. (See the section "Customizing HTML Pages Using SSIs" later in this chapter for a description of how to use these commands).

In addition to the two SSI commands, the Cisco IOS software supports several SSI variables defined for customizing HTML pages. SSI variables are used with the SSI ECHO command. One SSI variable is defined for all Cisco IOS platforms (SERVER_NAME), and other SSI variables are specifically defined for ISDN, Frame Relay, and asynchronous serial platforms. The format and a description of all the available SSI variables are provided in Table 14. (See the section "Customizing HTML Pages Using SSIs" later in this chapter for a description of how to use these SSI variables with the SSI ECHO command).

The SSI EXEC command is supported on all platforms. The SSI ECHO command, used with SSI variables, is supported on all platforms listed in Table 14.

Table 14 Description of SSI Variables

HTML Format of SSI Variable	Description of Variable Displayed on Browser Page	Cisco IOS Platforms This SSI Is Supported On
SERVER_NAME	Host name of the HTTP server.	All Cisco IOS platforms
EZSETUP_PASSWORD	Enable password (currently left blank).	Cisco 1000 series
EZSETUP_PASSWORD_VERIFY	Repeat of the enable password to verify accuracy (currently left blank).	Cisco 1000 series
EZSETUP_ETHERNET0_ADDRESS	IP address of the Ethernet interface 0.	Cisco 1000 series
EZSETUP_ETHERNET0_MASK	IP mask of the Ethernet interface 0.	Cisco 1000 series
EZSETUP_DNS_ADDRESS	Domain Name System (DNS) address used by the router.	Cisco 1000 series
EZSETUP_STANDARD_DEBUG_Y	Standard debug variable. Returns CHECKED if set to TRUE; otherwise, it is blank.	Cisco 1000 series
EZSETUP_STANDARD_DEBUG_N	Standard debug variable. Returns CHECKED if set to FALSE; otherwise, it is blank.	Cisco 1000 series
EZSETUP_ISDN_SWITCHTYPE	ISDN switch type.	Cisco 1003 and Cisco 1004
EZSETUP_ISDN_REMOTE_NAME	Name of remote ISDN system.	Cisco 1003 and Cisco 1004
EZSETUP_ISDN_REMOTE_NUMBER	Phone number of remote ISDN system.	Cisco 1003 and Cisco 1004
EZSETUP_ISDN_CHAP_PASSWORD	CHAP password of remote ISDN system.	Cisco 1003 and Cisco 1004
EZSETUP_ISDN_SPID1	ISDN SPID 1.	Cisco 1003 and Cisco 1004

Table 14 Description of SSI Variables (continued)

HTML Format of SSI Variable	Description of Variable Displayed on Browser Page	Cisco IOS Platforms This SSI Is Supported On
EZSETUP_ISDN_SPID2	ISDN SPID 2.	Cisco 1003 and Cisco 1004
EZSETUP_ISDN_SPEED_56	Speed of ISDN interface. Returns CHECKED if set to 56K; otherwise, it is blank.	Cisco 1003 and Cisco 1004
EZSETUP_ISDN_SPEED_64	Speed of ISDN interface. Returns CHECKED if set to 64K; otherwise, it is blank.	Cisco 1003 and Cisco 1004
EZSETUP_FR_ADDRESS	Frame Relay IP address.	Cisco 1005
EZSETUP_FR_MASK	Frame Relay IP mask.	Cisco 1005
EZSETUP_FR_DLCI	Frame Relay DLCI.	Cisco 1005
EZSETUP_ASYNC_REMOTE_NAME	Name of remote system.	Cisco 1005
EZSETUP_ASYNC_REMOTE_NUMBER	Phone number of remote system.	Cisco 1005
EZSETUP_ASYNC_CHAP_PASSWORD	CHAP password for remote system.	Cisco 1005
EZSETUP_ASYNC_LINE_PASSWORD	Async line password.	Cisco 1005
EZSETUP_ASYNC_MODEM_SPEED	Speed of async modem (either 14.4K or 28.8K).	Cisco 1005
EZSETUP_ASYNC_MODEM_SPEED_144K	Returns CHECKED if async modem speed is 14.4K; otherwise it is blank.	Cisco 1005
EZSETUP_ASYNC_MODEM_SPEED_288K	Returns CHECKED if async modem speed is 28.8K; otherwise it is blank.	Cisco 1005

Once you have designed a set of HTML pages that include SSIs, you can copy these pages to a Cisco IOS platform's Flash memory. When you retrieve these pages from Flash memory and display them using a Web browser, any SSI command that was designed into these pages will display either Cisco IOS command output or a current variable or identifier defined in Table 14. For example, the SSI ECHO command with the variable SERVER_NAME will display the current host name of the HTTP server you are using, and the SSI ECHO command with the variable EZSETUP_ISDN_SWITCHTYPE will display the current ISDN switch type you are using.

Using SSIs, you can customize set of HTML pages to appear in languages other than English and copy these pages to Flash memory on multiple Cisco IOS platforms. When you retrieve these pages from the Flash memory of a Cisco IOS platform, current variables and identifiers associated with the platform you are currently using are displayed. SSIs save you from needing to duplicate these international pages (considered relatively large images that contain 8-bit or multibyte characters) and store them in the source code for each platform you are using.

Customizing HTML Pages Using SSIs

When you are customizing an HTML page for a Web browser, type **<!--#exec cmd="*xxx*"-->** in your HTML file where you want Cisco IOS command output to appear on the browser page. Replace the *xxx variable* with any Cisco IOS EXEC mode command.

When you are customizing an HTML page for a Web browser, type **<!--#echo var="*yyy*"-->** in your HTML file where you want a value or identifier associated with a particular Cisco IOS platform (for example, an ISDN or Frame Relay platform) to appear on the browser page. Replace the *yyy* variable with an SSI variable described in Table 14.

Copying HTML Pages to Flash Memory

Once you have customized HTML pages using SSIs, copy your HTML pages to a Cisco IOS platform's Flash memory. To do this, save your pages using a filename appended with ".shtml" (for example, *filename*.shtml) and copy your file to Flash memory using a **copy** EXEC command (for example, the **copy tftp flash** command). (Refer to the Cisco IOS command references for a **copy** command compatible with your platform.)

Displaying HTML Files Containing SSIs

Once the Cisco Web browser UI is enabled, you can retrieve your HTML page from Flash memory and display it on the Cisco Web browser by typing **http://***router***/flash/***filename* in the URL window. Replace *router* with the host name or IP address of the current Cisco IOS platform you are using, and replace *filename* with the name of the file you created with ".shtml" appended, for example, http://myrouter/flash/ssi_file.shtml.

Cisco Web Browser UI Customization Examples

This section provides the following examples:

- Using the SSI EXEC Command Example
- Using the SSI ECHO Command Example

Using the SSI EXEC Command Example

The following example shows how the HTML SSI EXEC command can be used to execute a command. In this example, the Cisco IOS **show users** EXEC command is executed.

The contents of the HTML file in Flash memory are as follows:

```
<HTML>
<HEAD>
<TITLE> SSI EXEC Command Example</TITLE>
</HEAD>
<BODY>
This is an example of the SSI EXEC command
<HR>
<PRE>
<!--#exec cmd="show users"-->
</PRE>
<BR>
</BODY>
</HTML>
```

The contents that the Web browser receives when the HTML file is retrieved from Flash memory are as follows:

```
<HTML>
<HEAD>
<TITLE> SSI EXEC Command Example</TITLE>
</HEAD>
<BODY>
This is an example of the SSI EXEC command
<HR>
USERS:<BR>
<PRE>

Line    User  Host(s) Idle  Location
0 con 0       idle      12
2 vty 0       idle       0  router.cisco.com

</PRE>
<BR>
</BODY>
</HTML>
```

The Web browser shows the following text:

```
This is an example of the SSI EXEC command
------------------------------------------
USERS:
Line    User  Host(s) Idle  Location
0 con 0       idle      12
2 vty 0       idle       0  router.cisco.com
```

Using the SSI ECHO Command Example

The following is an example of the HTML SSI ECHO command used with the SSI variable *SERVER_NAME* (see Table 5) to display the Cisco IOS platform host name "rain."

The contents of the HTML file in Flash memory is as follows:

```
<HTML>
<HEAD>
<TITLE>SSI Echo Command Example</TITLE>
</HEAD>
<BODY>
This is an example of the SSI echo command
<HR>
The name of this server is:<BR>
<!--#echo var="SERVER_NAME"-->
<BR>
</BODY>
</HTML>
```

The contents that the Web browser receives when the HTML file is retrieved from Flash memory are as follows:

```
<HTML>
<HEAD>
<TITLE>SSI Echo Command Example</TITLE>
</HEAD>
<BODY>
This is an example of the SSI echo command
<HR>
The name of this server is:<BR>
rain
<BR>
</BODY>
</HTML>
```

The Web Browser shows the following text:

```
This is an example of the SSI echo command
------------------------------------------
The name of this server is:
rain
```

Using the Cisco IOS Integrated File System

Using the Cisco IOS Integrated File System

This chapter describes the Cisco IOS File System (IFS) feature, which provides a single interface to all the file systems available on your routing device, including the following:

- Flash memory file systems
- Network file systems (TFTP, rcp, and FTP)
- Any other endpoint for reading or writing data (such as NVRAM, the running configuration, ROM, raw system memory, system bundled microcode, Xmodem, Flash load helper log, modems, and BRI multiplexing device [mux] interfaces)

For a complete description of the IFS commands in this chapter, refer to the "Cisco IOS File System Commands" chapter in the "File Management Commands" part of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

IFS Use and Management Task List

This chapter describes the tasks you can perform to manage files using the Cisco IFS. Information about the IFS and its optional file management tasks are described in the following sections:

- Understanding IFS
- Copying Files Using URLs
- Using URLs in Commands
- Managing File Systems
- Flash Memory File System Types
- Remote File System Management
- NVRAM File System Management
- System File System Management

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Understanding IFS

IFS capabilities and benefits are described in the following sections:

- Displaying and Classifying Files
- Platform-Independent Commands
- Minimal Prompting for Commands
- Creating and Navigating Directories

Displaying and Classifying Files

With IFS, all files can be viewed and classified (image, text file, and so on), including files on remote servers. For example, you may want to determine the size and type of an image on a remote server before you copy it to ensure that it is a valid image. You can also display a configuration file on a remote server to verify that it is the correct configuration file before you load the file on the router.

Platform-Independent Commands

With IFS, the file system user interface is no longer platform-specific. Commands have the same syntax, regardless of which platform is used. Thus, you can use the same commands for all of your routers.

However, not all commands are supported on all platforms and file systems. Because different types of file systems support different operations, certain commands are not available for all file systems. Platforms will support commands for the file systems they use.

Minimal Prompting for Commands

IFS minimizes the required prompting for many commands, such as the **copy** EXEC command. You can enter all of the required information in the command line, rather than needing to provide information when the system prompts you for it. For example, if you want to copy a file to an FTP server, on a single line you can specify the specific location on the router of the source file, the specific location of the destination file on the FTP server, and the username and password to use when connecting to the FTP server. However, to have the router prompt you for the needed information, you can still enter the minimal form of the command.

Depending on the current configuration of the **file prompt** global configuration command and the type of command you entered, the router may prompt you for confirmation, even if you have provided all the information in the command. In these cases, the default value will be the value entered in the command. Press Return to confirm the values.

Creating and Navigating Directories

With IFS, you can navigate to different directories and list the files in a directory. On newer platforms, you can create subdirectories in Flash memory or on a disk.

Copying Files Using URLs

The new file system interface uses Uniform Resource Locators (URLs) to specify the location of a file. URLs are commonly used to specify files or locations on the World Wide Web. However, on Cisco routers, they can now be used to specify the location of files on the router or remote file servers.

On Cisco routers, use URLs in commands to specify the location of the file or directory. For example, if you want to copy a file from one location to another, use the **copy** *source-url destination-url* EXEC command.

The format of URLs used by the routers can vary from the format you may be used to using. There are also a variety of formats that can be used, based on the location of the file.

Information for copying files using URLs is included in the following sections:

- Specifying Files on a Network Server
- Specifying Local Files
- Using URL Prefixes

Specifying Files on a Network Server

To specify a file on a network server, use one of the following forms:

- **ftp:**[[//[*username*[:*password*]@]*location*]/*directory*]/*filename*
- **rcp:**[[//[*username*@]*location*]/*directory*]/*filename*
- **tftp:**[[//*location*]/*directory*]/*filename*

The *location* can be an IP address or a host name. The *username* variable, if specified, overrides the username specified by the **ip rcmd remote-username** or **ip ftp username** global configuration command. The *password* overrides the password specified by the **ip ftp password** global configuration command.

The file path (directory and filename) is specified relative to the directory used for file transfers. For example, on UNIX file servers, TFTP pathnames start in the /tftpboot directory, and rcp and FTP paths start in the home directory associated with the username.

The following example specifies the file named c7200-j-mz.112-current on the TFTP server named myserver.cisco.com. The file is located in the directory named /tftpboot/master.

```
tftp://myserver.cisco.com/master/c7200-j-mz.112-current
```

The following example specifies the file named mill-config on the server named enterprise.cisco.com. The router uses the username liberty and the password secret to access this server via FTP.

```
ftp://liberty:secret@enterprise.cisco.com/mill-config
```

Specifying Local Files

Use the *prefix***:**[*directory***/**]*filename* syntax to specify a file located on the router. You can use this form to specify a file in Flash memory or NVRAM.

For example, `nvram:startup-config` specifies the startup configuration in NVRAM, and `flash:configs/backup-config` specifies the file named backup-config in the configs directory of Flash memory.

When referring to a file system instead of a file, use the *prefix***:** form. This form specifies the file system itself, rather than a file in the file system. Use this form to issue commands on file systems themselves, such as commands to list the files in a file system or to format the file system.

For example, slot0: can indicate the first Personal Computer Memory Card Industry Association (PCMCIA) Flash memory card in slot 0.

Using URL Prefixes

The URL prefix specifies the file system. The list of available file systems differs by platform and operation. Refer to your product documentation or use the **show file systems** EXEC command to determine which prefixes are available on your platform. File system prefixes are listed in Table 15.

Table 15 File System Prefixes

Prefix	File System
bootflash:	Boot Flash memory.
disk0:	Rotating media.
flash:	Flash memory. This prefix is available on all platforms. For platforms that do not have a device named flash:, the prefix flash: is aliased to slot0:. Therefore, you can use the prefix flash: to refer to the main Flash memory storage area on all platforms.
flh:	Flash load helper log files.
ftp:	FTP network server.
null:	Null destination for copies. You can copy a remote file to null to determine its size.
nvram:	NVRAM.
rcp:	Remote copy protocol network server.
slavebootflash:	Internal Flash memory on a slave RSP card of a router configured for high system availability (HSA).
slavenvram:	NVRAM on a slave Route/Switch Processor (RSP) card of a router configured for HSA.
slaveslot0:	First PCMCIA card on a slave RSP card of a router configured for HSA.
slaveslot1:	Second PCMCIA card on a slave RSP card of a router configured for HSA.
slot0:	First PCMCIA Flash memory card.
slot1:	Second PCMCIA Flash memory card.
system:	Contains the system memory, including the running configuration.
tftp:	TFTP network server.

Table 15 File System Prefixes (continued)

Prefix	File System
xmodem:	Obtain the file from a network machine using the Xmodem protocol.
ymodem:	Obtain the file from a network machine using the Ymodem protocol.

Note Maintenance Operation Protocol (MOP) servers are no longer supported as file systems.

In all commands, the colon is required after the file system name. However, commands that did not require the colon previously will continue to be supported, although they will not be available in the context-sensitive help.

URL Prefix for Partitioned Devices

For partitioned devices, the URL prefix includes the partition number. The syntax is *device***:***partition-number***:** for the prefix on a partitioned device.

For example, flash:2: refers to the second partition in Flash memory.

URL Component Lengths

Table 16 lists the maximum lengths in characters of the different URL components.

Table 16 URL Component Lengths

Component	Length (Number of Characters)
Prefix	31
Username	15
Password	15
Hostname	31
Directory	63
Filename	63

Using URLs in Commands

Depending on which command you are using, different file systems are available. Some file systems can only serve as a source for files, not a destination. For example, you cannot copy to another machine using Xmodem. Other operations, such as **format** and **erase**, are only supported by certain file systems on certain platforms.

The following sections describe the use of for using URLs in commands:

- Determining File Systems Supporting a Command
- Using the Default File System

- Using Tab Completion
- Listing Files in a File System

Determining File Systems Supporting a Command

Use the context-sensitive help to determine which file systems can be used for a particular command. In the following example, the context-sensitive help displays which file systems can be used as sources for the **copy** EXEC command. The output will vary based on the platform.

```
Router# copy ?
  /erase      Erase destination file system.
  bootflash:  Copy from bootflash: file system
  flash:      Copy from flash: file system
  ftp:        Copy from ftp: file system
  null:       Copy from null: file system
  nvram:      Copy from nvram: file system
  rcp:        Copy from rcp: file system
  system:     Copy from system: file system
  tftp:       Copy from tftp: file system
```

Using the Default File System

For most commands, if no file system is specified, the file is assumed to be in the default directory, as specified by the **cd** command.

```
Router# pwd
slot0:
Router# dir
Directory of slot0:/

  1  -rw-     4720148   Aug 29 1997 17:49:36  hampton/nitro/c7200-j-mz
  2  -rw-     4767328   Oct 01 1997 18:42:53  c7200-js-mz
  5  -rw-         639   Oct 02 1997 12:09:32  foo
  7  -rw-         639   Oct 02 1997 12:37:13  the_time

20578304 bytes total (3104544 bytes free)
Router# cd nvram:
Router# dir
Directory of nvram:/

  1  -rw-        2725              <no date>  startup-config
  2  ----           0              <no date>  private-config
  3  -rw-        2725              <no date>  underlying-config

129016 bytes total (126291 bytes free)
```

Using Tab Completion

You can use tab completion to reduce the number of characters you need to type for a command. Type the first few characters of the filename, and press the Tab key. If the characters are unique to a filename, the router will complete the filename for you. Continue entering the command as normal and press Return to execute the command.

In the following example, the router completes the filename startup-config because it is the only file in the nvram: file system that starts with “s”:

```
Router# show file info nvram:s<tab>
Router# show file info nvram:startup-config<Enter>
```

If you use tab completion without specifying any characters, the router uses the first file in the file system.

```
Router# show file info nvram:<tab>
Router# show file info nvram:private-config<Enter>
```

Listing Files in a File System

For many commands, you can get a listing of the files in a file system on the router by using the context-sensitive help. In the following example, the router lists the files in NVRAM:

```
Router# show file info nvram:?
nvram:private-config  nvram:startup-config  nvram:underlying-config
```

Managing File Systems

To manage file systems, perform the tasks described in the following sections.

- Listing Available File Systems
- Setting the Default File System
- Displaying the Current Default File System
- Displaying Information About Files on a File System
- Displaying a File

Listing Available File Systems

Not all file systems are supported on every platform. To list the file systems available on your platform, use the following EXEC mode command:

Command	Purpose
`Router> show file systems`	Lists the file systems available on your platform. This command also displays information about each file system.

Setting the Default File System

You can specify the file system or directory that the system uses as the default file system. Setting the default file system allows you to omit an optional *filesystem***:** argument from related commands. For all EXEC commands that have an optional *filesystem***:** argument, the system uses the file system specified by the **cd** EXEC command when you omit the optional *filesystem***:** argument. For example, the **dir** EXEC command contains an optional *filesystem***:** argument and displays a list of files on the file system.

To set a default file system, use the following command in EXEC mode:

Command	Purpose
Router> **cd** *filesystem*:	Sets a default Flash memory device.

The following example sets the default file system to the Flash memory card inserted in slot 0:

```
cd slot0:
```

Displaying the Current Default File System

To display the current default file system, as specified by the **cd** EXEC command, use the following command in EXEC mode:

Command	Purpose
Router> **pwd**	Displays the current file system.

The following example shows that the default file system is slot 0:

```
Router> pwd
slot0:
```

The following example uses the **cd** command to change the default file system to system and then uses the **pwd** command to verify that the default file system was changed:

```
Router> cd system:
Router> pwd
system:
```

Displaying Information About Files on a File System

You can display a list of the contents of a file system before manipulating its contents. For example, before copying a new configuration file to Flash memory, you may want to verify that the file system does not already contain a configuration file with the same name. Similarly, before copying a Flash configuration file to another location, you may want to verify its filename for use in another command.

To display information about files on a file system, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# **dir** [**/all**] [*filesystem*:][*filename*]	Displays a list of files on a file system.
Router# **show file systems**	Displays detailed information about each of the files on a file system.
Router# **show file information** *file-url*	Displays information about a specific file.
Router# **show file descriptors**	Displays a list of open file descriptors.

The following example compares the different commands used to display information about files for the PCMCIA card in the first slot. Notice that deleted files appear in the **dir /all** command output but not in the **dir** command output.

```
Router# dir slot0:
Directory of slot0:/

  1  -rw-     4720148   Aug 29 1997 17:49:36  hampton/nitro/c7200-j-mz
  2  -rw-     4767328   Oct 01 1997 18:42:53  c7200-js-mz
  5  -rw-         639   Oct 02 1997 12:09:32  foo
  7  -rw-         639   Oct 02 1997 12:37:13  the_time

20578304 bytes total (3104544 bytes free)

Router# dir /all slot0:
Directory of slot0:/

  1  -rw-     4720148   Aug 29 1997 17:49:36  hampton/nitro/c7200-j-mz
  2  -rw-     4767328   Oct 01 1997 18:42:53  c7200-js-mz
  3  -rw-     7982828   Oct 01 1997 18:48:14  [rsp-jsv-mz]
  4  -rw-         639   Oct 02 1997 12:09:17  [the_time]
  5  -rw-         639   Oct 02 1997 12:09:32  foo
  6  -rw-         639   Oct 02 1997 12:37:01  [the_time]
  7  -rw-         639   Oct 02 1997 12:37:13  the_time

20578304 bytes total (3104544 bytes free)

Router# show slot0:
-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time------ name
1   .. unknown  317FBA1B  4A0694   24  4720148 Aug 29 1997 17:49:36 hampton/nitz
2   .. unknown  9237F3FF  92C574   11  4767328 Oct 01 1997 18:42:53 c7200-js-mz
3   .D unknown  71AB01F1 10C94E0   10  7982828 Oct 01 1997 18:48:14 rsp-jsv-mz
4   .D unknown  96DACD45 10C97E0    8      639 Oct 02 1997 12:09:17 the_time
5   .. unknown  96DACD45 10C9AE0    3      639 Oct 02 1997 12:09:32 foo
6   .D unknown  96DACD45 10C9DE0    8      639 Oct 02 1997 12:37:01 the_time
7   .. unknown  96DACD45 10CA0E0    8      639 Oct 02 1997 12:37:13 the_time

3104544 bytes available (17473760 bytes used)
```

Displaying a File

To display the contents of any readable file, including a file on a remote file system, use the following command in EXEC mode:

Command	Purpose
`Router#` **`more`** `[`**`/ascii`** `|` **`/binary`** `|` **`/ebcdic`**`]` *`file-url`*	Displays the specified file.

The following example displays the contents of a configuration file on a TFTP server:

```
Router# more tftp://serverA/hampton/savedconfig

!
! Saved configuration on server
!
version 11.3
service timestamps log datetime localtime
service linenumber
service udp-small-servers
service pt-vty-logging
!
end
```

Flash Memory File System Types

Cisco platforms use one of the following three different Flash memory file system types:

- Class A Flash File Systems
- Class B Flash File Systems
- Class C Flash File Systems

The methods used for erasing, deleting, and recovering files depend on the class of the Flash file system. Some commands are supported on only one or two file system types. The command reference documentation notes commands that are not supported on all file system types.

See Table 17 to determine which Flash memory file system type your platform uses.

Table 17 Flash Memory File System Types

Type	Platforms
Class A	Cisco 7000 series (including the Cisco 7500 series), Cisco 12000 Gigabit Switch Router (GSR), LS1010
Class B	Cisco 1003, Cisco 1004, Cisco 1005, Cisco 2500 series, Cisco 3600 series, Cisco 4000 series, Cisco AS5200
Class C	Cisco MC3810, disk0 of SC3640

Class A Flash File Systems

On Class A Flash file systems, you can delete individual files using the **delete** EXEC command and later recover these files with the **undelete** EXEC command. The **delete** command marks the files as "deleted," but the files still take up space in Flash memory. To permanently delete the files, use the **squeeze** EXEC command. The **squeeze** command removes all of the files marked "deleted" from the specified Flash memory device. These files can no longer be recovered. To erase all of the files on a Flash device, use the **format** EXEC command.

Deleting Files on a Flash Memory Device

When you no longer need a file on a Flash memory device, you can delete it. When you delete a file, the router simply marks the file as deleted, but it does not erase the file. This feature allows you to recover a deleted file, as discussed in the following section. You may want to recover a "deleted" image or configuration file if the new image or configuration file becomes corrupted.

To delete a file from a specified Flash memory device, use the following EXEC mode command:

Command	Purpose
`Router#` **`delete`** `[`*`device`*`:]`*`filename`*	Deletes a file from a Flash memory device.

If you omit the device, the router uses the default device specified by the **cd** EXEC command.

If you attempt to delete the file specified by the CONFIG_FILE or BOOTLDR environment variable, the system prompts you to confirm the deletion. Also, if you attempt to delete the last valid system image specified in the BOOT environment variable, the system prompts you to confirm the deletion.

The following example deletes the file named myconfig from a Flash memory card inserted in slot 0:

```
delete slot0:myconfig
```

Recovering Deleted Files on a Flash Memory Device

You can undelete a deleted file. For example, you may want to revert to a previous configuration file because the current one is corrupt.

To undelete a deleted file on a Flash memory device, use the following commands in EXEC mode:

	Command	Purpose
Step 1	`Router# dir /all [filesystem:]`	Determines the index of the deleted file.
Step 2	`Router# undelete index [filesystem:]`	Restores a deleted file on a Flash memory device.

You must undelete a file by its index because you can have multiple deleted files with the same name. For example, the "deleted" list could contain multiple configuration files with the name router-config. You undelete by index to indicate which of the many router-config files from the list to undelete. Use the **dir** command with the **/all** option to learn the index number of the file you want to undelete.

You cannot undelete a file if a valid file with the same name exists. Instead, first delete the existing file and then undelete the file you want. For example, if you had a file with the name router-config and you wanted to use a file with the same name that you had previously deleted, you cannot simply undelete the previous version by index. You must first delete the existing router-config file and then undelete the previous router-config file by index. You can undelete a file as long as the file has not been permanently erased with the **squeeze** EXEC command. You can delete and undelete a file up to 15 times.

The following example recovers the deleted file whose index number is 1 to the Flash memory card inserted in slot 0:

```
undelete 1 slot0:
```

Permanently Deleting Files on a Flash Memory Device

When a Flash memory device is full, you may need to rearrange the files so that the space used by the deleted files can be reclaimed. To determine whether a Flash memory device is full, use the **dir** EXEC command.

To permanently delete files on a Flash memory device, use the following command in privileged EXEC mode:

Command	Purpose
`Router# squeeze filesystem:`	Permanently deletes all files marked "deleted" on a Flash memory device.

On Cisco 2600 and 3600 series routers,the entire flash file system needs to be erased once before the **squeeze** command can be used. After being erased once, the squeeze command should operate properly on the flash file system for the rest of the flash file system's history.

To erase an entire flash file system on a Cisco 2600 or 3600 series router, perform the following steps:

Command	Purpose
`Router# no partition flash-filesystem:`	Removes all partitions on the specified flash file system. **Note** The reason for removing partitions is to ensure that the entire flash file system is erased. The **squeeze** command can be used in a flash file system with partitions after the flash file system is erased once.
`Router# erase filesystem:`	Erases all of the file on the specified flash file system.

When you issue the **squeeze** command, the router copies all valid files to the beginning of Flash memory and erases all files marked "deleted." At this point, you cannot recover deleted files, and you can now write to the reclaimed Flash memory space.

Note The squeeze operation can take as long as several minutes because it can involve erasing and rewriting almost an entire Flash memory space.

Verifying Flash

To recompute and verify the checksum of a file in Flash memory on a Class A Flash file system, use the **verify** EXEC command.

Deleting and Recovering a Class A Flash File System Example

In the following example, the image named c7200-js-mz is deleted and undeleted. Note that the deleted file does not appear in the output for the first **dir** EXEC command, but it appears in the output for the **dir /all** EXEC command.

```
Router# delete slot1:
Delete filename []? c7200-js-mz
Delete slot1:c7200-js-mz? [confirm]
Router# dir slot1:
Directory of slot1:/

No such file

20578304 bytes total (15754684 bytes free)
Router# dir /all slot1:
Directory of slot1:/

  1  -rw-     4823492   Dec 17 1997 13:21:53  [c7200-js-mz]

20578304 bytes total (15754684 bytes free)
Router# undelete 1 slot1:
Router# dir slot1:
Directory of slot1:/

  1  -rw-     4823492   Dec 17 1997 13:21:53 c7200-js-mz

20578304 bytes total (15754684 bytes free)
```

In the following example, the image is deleted. In order to reclaim the space taken up by the deleted file, the **squeeze** EXEC command is issued.

```
Router# delete slot1:c7200-js-mz
Delete filename [c7200-js-mz]?
Delete slot1:c7200-js-mz? [confirm]
Router# squeeze slot1:
All deleted files will be removed. Continue? [confirm]
Squeeze operation may take a while. Continue? [confirm]
Erasing squeeze log
Squeeze of slot1: complete
Router# dir /all slot1:
Directory of slot1:/

No such file

20578304 bytes total (20578304 bytes free)
```

Class B Flash File Systems

On Class B Flash file systems, you can delete individual files with the **delete** EXEC command. The **delete** command marks the file as "deleted." The file is still present in Flash memory and takes up space. To recover the file, use the **undelete** EXEC command. To reclaim any space in Flash memory, you must erase the entire Flash file system with the **erase** EXEC command.

Deleting Files on a Flash Memory Device

When you no longer need a file on a Flash memory device, you can delete it. When you delete a file, the router simply marks the file as deleted, but it does not erase the file. This feature allows you to recover a deleted file, as discussed in the following section. You may want to recover a "deleted" image or configuration file if the new image or configuration file becomes corrupted.

To delete a file from a specified Flash memory device, use the following EXEC mode command:

Command	Purpose
Router# **delete** [*device*:]*filename*	Deletes a file from a Flash memory device.

If you omit the device, the router uses the default device specified by the **cd** EXEC command.

The following example deletes the file named myconfig from a Flash memory card inserted in slot 0:

```
delete slot0:myconfig
```

Recovering Deleted Files on a Flash Memory Device

You can undelete a deleted file. For example, you may want to revert to a previous configuration file because the current one is corrupt.

To undelete a deleted file on a Flash memory device, use the following EXEC mode commands:

	Command	Purpose
Step 1	Router# **dir /all** [*filesystem*:]	Determines the index of the deleted file.
Step 2	Router# **undelete** *index* [*filesystem*:]	Undeletes a deleted file on a Flash memory device.

You must undelete a file by its index because you can have multiple deleted files with the same name. For example, the "deleted" list could contain multiple configuration files with the name router-config. You undelete by index to indicate which of the many router-config files from the list to undelete. Use the **dir** command with the **/all** option to learn the index number of the file you want to undelete.

You cannot undelete a file if a valid (undeleted) one with the same name exists. Instead, first delete the existing file and then undelete the file you want. For example, if you had an undeleted version of the router-config file and you wanted to use a previous, deleted version instead, you cannot simply undelete the previous version by index. You must first delete the existing router-config file and then undelete the previous router-config file by index. You can undelete a file as long as the file system has not been permanently erased with the **erase** EXEC command. You can delete and undelete a file up to 15 times.

The following example recovers the deleted file whose index number is 1 to the Flash memory card inserted in slot 0:

```
undelete 1 slot0:
```

Erasing Flash Memory

In order to reclaim any space taken up by files in Flash memory, you must erase the entire file system using the **erase flash:** or **erase bootflash:** EXEC command. These commands reclaim all of the space in Flash memory, erasing all files, deleted or not, in the process. Once erased, these files cannot be recovered. Before erasing Flash memory, save any files you want to keep in another location (an FTP server, for example). Copy the files back to Flash memory after you have erased the device.

To erase a Flash memory device, use the following command in EXEC mode:

Command	Purpose
`Router# erase filesystem:`	Erases the Flash file system.

Erasing a File System Example

The following example erases all files in the second partition in Flash memory:

```
Router# erase flash:2

System flash directory, partition 2:
File  Length   Name/status
  1   1711088  dirt/gate/c1600-i-mz
[1711152 bytes used, 15066064 available, 16777216 total]

Erase flash device, partition 2? [confirm]
Are you sure? [yes/no]: yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
```

Verifying Flash

To recompute and verify the checksum of a file in Flash memory on a Class B Flash file system, use the **verify** EXEC command.

Class C Flash File Systems

On Class C Flash memory file systems, you can delete individual files with the **delete** EXEC command. Files cannot be reclaimed once they have been deleted. Instead, the Flash file system space is reclaimed dynamically. To erase all of the files in Flash, use the **format** EXEC command.

Deleting Files on a Flash Memory Device

When you no longer need a file on a Flash memory device, you can delete it. When you delete a file on a Class C file system, the file is deleted permanently. The router reclaims the space dynamically.

To delete a file from a specified Flash device, use the following command in EXEC mode:

Command	Purpose
`Router#` **`delete`** `[`*`device`*`:]`*`filename`*	Deletes a file from a Flash memory device.

If you omit the device, the router uses the default device specified by the **cd** EXEC command.

If you attempt to delete the file specified by the CONFIG_FILE or BOOTLDR environment variable, the system prompts you to confirm the deletion. Also, if you attempt to delete the last valid system image specified in the BOOT environment variable, the system prompts you to confirm the deletion.

The following example permanently deletes the file named myconfig from a Flash memory card inserted in slot 0:

```
delete slot0:myconfig
```

Formatting Flash

To format a Class C Flash file system, use the following command in EXEC mode:

Command	Purpose
`Router#` **`format`** *`filesystem`*	Formats a Flash file system.

If you format a Flash device, all of the files are erased and cannot be recovered.

Creating and Removing Directories

On Class C Flash file systems, you can create a new directory with the **mkdir** EXEC command. To remove a directory from a Flash file system, use the **rmdir** EXEC command.

On Class C Flash file systems, you can rename a file using the **rename** EXEC command.

Checking Flash File Systems

On Class C Flash file systems, you can check a file system for damage and repair any problems using the **fsck** EXEC command.

Remote File System Management

On remote file systems (file systems on FTP, rcp, or TFTP servers) you can perform the following tasks:

- View the contents of a file with the **more** EXEC command.
- Copy files to or from the router using the **copy** EXEC command.
- Display information about a file using the **show file information** EXEC command.

Note You cannot delete files on remote systems.

NVRAM File System Management

On most platforms, NVRAM contains the startup configuration. On Class A Flash file system platforms, the CONFIG_FILE environment variable specifies the location of the startup configuration. However, the file URL nvram:startup-config always specifies the startup configuration, regardless of the CONFIG_FILE environment variable.

You can display the startup-config (with the **more nvram:startup-config** EXEC command), replace the startup config with a new configuration file (with the **copy** *source-url* **nvram:startup-config** EXEC command), save the startup configuration to another location (with the **copy nvram:startup-config** *destination-url* EXEC command), and erase the contents of NVRAM (with the **erase nvram:** EXEC command). The **erase nvram:** command also deletes the startup configuration if another location is specified by the CONFIG_FILE variable.

The following example displays the startup configuration:

```
nnm3640-2# more nvram:startup-config
Using 2279 out of 129016 bytes
!
! Last configuration change at 10:57:25 PST Wed Apr 22 1998
! NVRAM config last updated at 10:57:27 PST Wed Apr 22 1998
!
version 11.3
service timestamps log datetime localtime
service linenumber
service udp-small-servers
service pt-vty-logging
...
end
```

The following example displays the contents of the NVRAM file system on a Class A Flash file system platform. The file named startup-config is the current startup configuration file, in physical NVRAM or in Flash memory. If the file is located in a Flash memory file system, this entry is a symbolic link to the actual file. The file named underlying-config is always the NVRAM version of the configuration.

```
Router# dir nvram:
  Directory of nvram:/

    1  -rw-        2703              <no date>  startup-config
    2  ----           5              <no date>  private-config
    3  -rw-        2703              <no date>  underlying-config

  129016 bytes total (126313 bytes free)
```

System File System Management

The “system” file system contains the system memory and the current running configuration. You can display the current configuration (with the **show running-config** or **more system:running-config** EXEC command), save the current configuration to another location (with the **copy system:running-config** *destination-url* EXEC command), and add configuration commands to the current configuration (with the **copy** *source-url* **system:running-config** EXEC command).

The following example changes to the “system” file system, displays the contents of the file system, and displays the running configuration:

```
Router# cd ?
  bootflash:  Directory name
  flash:      Directory name
  lex:        Directory name
  modem:      Directory name
  null:       Directory name
  nvram:      Directory name
  system:     Directory name
  vfc:        Directory name
  <cr>

Router# cd system:?
system:memory  system:running-config  system:ucode  system:vfiles

Router# cd system:
Router# dir
Directory of system:/

    6  dr-x           0              <no date>  memory
    1  -rw-        7786   Apr 22 2001 03:41:39  running-config

No space information available

nnm3640-2# more system:running-config
!
! No configuration change since last restart
!
version 12.2
service timestamps log datetime localtime
service linenumber
service udp-small-servers
service pt-vty-logging
!
.
.
.
end
```

On some platforms, the system file system contains microcode in its ucode directory, as follows:

```
Router# dir system:/ucode
Directory of system:/ucode/

 21  -r--       22900              <no date>  aip20-13
 18  -r--       32724              <no date>  eip20-3
 25  -r--      123130              <no date>  feip20-6
 19  -r--       25610              <no date>  fip20-1
 22  -r--        7742              <no date>  fsip20-7
 23  -r--       17130              <no date>  hip20-1
 24  -r--       36450              <no date>  mip22-2
 29  -r--      154752              <no date>  posip20-0
```

```
 28  -r--      704688             <no date>  rsp220-0
 20  -r--       33529             <no date>  trip20-1
 26  -r--      939130             <no date>  vip22-20
 27  -r--     1107862             <no date>  vip222-20

No space information available
```

File System Check and Repair for PCMCIA ATA Disks

This feature introduces a File-System-Check (fsck) utility in Cisco IOS software for FAT filesystems on PCMCIA disks. The utility performs functions such as checking the boot sector and partition table, checking file and directory structure, reclaiming unused disk space, and updating the FAT file structure.

Feature Specifications for the File System Check and Repair for PCMCIA ATA Disks Feature

Feature History	
Release	**Modification**
12.2(13)T, 12.0(22)S	This feature was introduced.
Supported Platforms	
See Cisco Feature Navigator.	

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Contents

This document contains the following sections:

Information About File System Check and Repair

Prior to the introduction of the file system check (fsck) utility, corrupt files could not be removed from ATA disks using the Cisco IOS command-line interface CLI.

Files (or file metadata) in an ATA disk can be corrupted by a variety of events, from power failures or system crashes to simple tftp copy failures. Prior to the introduction of the file system check (fsck) utility, corrupted files could not be deleted from a usable ATA disk without removing, reformatting, and reinstalling the disk.

The **fsck** privileged EXEC command allows you to conveniently recover wasted disk space directly from the CLI.

How to Use the File System Check and Repair Feature

No configuration tasks are associated with this enhancement. For usage guidelines, see the "Command Reference" section on page 3.

Additional References

None.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and lots more. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **fsck**

USB Storage

The USB Storage feature enables certain models of Cisco routers to support USB flash modules and with SmartCard technology (which is owned by Aladdin Knowledge Systems) in a USB key form factor (also referred to as a USB eToken) to provide secure access to a router.

USB eTokens provides secure configuration distribution and allows users to store Virtual Private Network (VPN) credentials for deployment. USB flash drives allow users to store images and configurations external to the router.

Feature History for USB Storage

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for USB Storage

Before you can use a USB Flash module or an eToken, you should have the following system requirements:

- A Cisco 871 router, Cisco 1800 series, Cisco 2800 series, or a Cisco 3800 series router
- At least a Cisco IOS Release 12.3(14)T image running on any of the supported platforms
- A Cisco supported USB flash or USB eToken
- A k9 image is required for USB eToken support. (However, USB flash support is available in all images.)

Restrictions for USB Storage

- USB eToken support requires a 3DES (k9) Cisco IOS software image, which provides secure file storage.
- USB hubs are currently not supported. Thus, the number of supported devices is limited to the number of available USB ports on the router chassis.
- You cannot boot an image from an eToken or a USB flash. (However, you can boot a configuration from both an eToken and flash.)

Information About USB Storage

To use a USB flash module and a secure eToken on your router, you should understand the following concepts:

- Roles of the USB eToken and the USB Flash, page 2
- Benefits of USB Storage, page 4

Roles of the USB eToken and the USB Flash

Both USB eTokens and USB flash modules can be used to store files (such as router configurations). The following sections discuss how each device functions and describe the differences between each device:

- How a USB eToken Works, page 2
- How a USB Flash Works, page 3
- Functionality Differences Between an eToken and a USB Flash, page 3

How a USB eToken Works

A SmartCard is a small plastic card, containing a microprocessor and memory that allows you to store and process data. A SmartCard eToken is a SmartCard with a USB interface. The eToken can securely store any type of file within its available storage space (32KB). Configuration files that are stored on the eToken can be encrypted and accessed only via a user PIN. The router will not load the configuration file unless the proper PIN has been configured for secure deployment of router configuration files.

After you plug the eToken into the router, you must log into the eToken; thereafter, you can change default settings, such as the user PIN (default: 1234567890) and the allowed number of failed login attempts before future logins are refused (default: 15 attempts). For more information on accessing and configuring the eToken, see the section "Accessing and Setting Up the eToken."

After you have successfully logged into the eToken, you can copy files from the router on to the eToken via the **copy** command. By default, after the eToken is removed from the router, all associated RSA keys are removed; IPSec tunnels are not torn down until the next Internet Key Exchange (IKE) negotiation period. (To change the default behavior and configure a specified length of time before the IPSec tunnels are torn down, issue the **crypto pki token removal timeout** command.)

For more information about the eToken by Aladdin Knowledge Systems, see the Aladdin website at http://www.aladdin.com/etoken/cisco/.

How a USB Flash Works

A Cisco USB flash module allows you to store and deploy router configurations and Cisco IOS software images. Cisco USB flash modules are available in 64MB, 128 MB, and 256MB versions.

Note The USB flash is not a replacement for the router compact flash, which must be present for the router to boot.

After you plug the USB flash module into the router, the router will automatically begin to boot the configuration file if the start-up configuration contains the **boot config** command to specify the new configuration located on the USB flash device; for example **boot config usbflash0: new-config**.

Functionality Differences Between an eToken and a USB Flash

Both eTokens and USB flash provide users with secondary storage; however, each device has its own benefits and limitations. To help determine which device better suits your needs, Table 1 highlights the functionality differences between the eToken and the USB flash.

Table 1 Functionality Differences Between an eToken and a USB Flash

Function	USB eToken	USB Flash
Accessibility	Used to securely store and transfer digital certificates, preshared keys, and router configurations from the eToken to the router.	Used to store and deploy router configurations and images from the USB Flash to the router.
Storage Size	32KB	• 64MB • 128MB • 256MB
File Types	• Typically used to store digital certificates, preshared keys, and router configurations for IPSec VPNs. • eTokens cannot store Cisco IOS images.	Stores a file type that might be stored on a compact flash.

Table 1 Functionality Differences Between an eToken and a USB Flash (Continued)

Function	USB eToken	USB Flash
Security	• Files can be encrypted and accessed only with a user PIN. • Files can also be stored in a nonsecure format.	Files can be stored only in a nonsecure format.
Boot Configurations	• The router can use the configuration stored in the eToken during boot time • The router can use the secondary configuration stored in the eToken during boot time. (A secondary configuration allows users to load their IPSec configuration.)	• Configuration file can be automatically transferred from the USB Flash to the router if the **boot config** command is issued (for example, **boot config usbflash0: new-config**).

Benefits of USB Storage

USB flash drive and USB eToken support on a Cisco router provides the following application benefits:

Removable Credentials: Provide or Store VPN Credentials on an External Device for Deployment

An Aladdin eToken can use SmartCard technology to store a digital certificate and configuration for IPSec VPN deployment. This ability enhances the capability of the router to generate RSA public keys to authenticate at least one IPSec tunnel. (Because a router can initiate multiple IPSec tunnels, the eToken can contain several certificates, as appropriate.)

Storing VPN credentials on an external device reduces the threat of compromising secure data.

PIN Configuration for Secure File Deployment

An Aladdin eToken can store a configuration file that can be used for enabling encryption on the router via a user-configured PIN. (That is, no digital certificates, preshared keys, or VPNs are used.)

Touchless or Low Touch Configuration

Both the eToken and USB Flash can provide remote software configuration and provisioning with little or no human interaction. Configuration is set up as an automated process. That is, both devices can store a bootstrap configuration that the router can use to boot from after the eToken or USB Flash has been inserted into the router. The bootstrap configuration connects the router to a TFTP server, which contains a configuration that completely configures the router.

How to Set Up and Use USB Modules on Cisco Routers

This section contains the following procedures that allow you to configure a router to support USB modules:

- Storing the Configuration on an External USB Flash Drive or eToken, page 5
- Accessing and Setting Up the eToken, page 5
- Troubleshooting USB Flash Drives and eTokens, page 10

Storing the Configuration on an External USB Flash Drive or eToken

Use the following task to store the configuration file in the USB flash drive module or in an eToken.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. boot config {usbflash[0-9]:filename | usbtoken[0-9]:filename}

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**boot config** {**usbflash**[**0-9**]**:***filename* | **usbtoken**[**0-9**]**:***filename*} **Example:** `Router(config)# boot config usbflash0:`	Specifies that the startup configuration file is stored in a USB Flash drive or secure eToken. **Note** If a USB flash drive is used, the router will boot a boot helper from **flash:**. The boot helper is a Cisco IOS image that resides in **flash:**. The Cisco IOS image that is used must be USB-aware.

Accessing and Setting Up the eToken

After you have inserted the eToken into the Cisco router, you must log into the eToken as shown in the following task:

- Logging Into the eToken, page 6 (required)

After you have logged into the eToken, you can perform administrative tasks, such as changing the user PIN and copying files from the router to the eToken, as shown in the following task:

- Setting Administrative Functions on the eToken, page 7 (optional)

Use of RSA Keys with an eToken

- RSA keys are loaded after the eToken is successfully logged into the router.
- By default, newly generated RSA keys are stored on the most recently inserted eToken. Regenerated keys should be stored in the same location that the original RSA key was generated.

Logging Into the eToken

Use this task to log into an eToken manually or automatically.

Automatic Login

Automatic login allows the router to completely come back up without any user or operator intervention. The PIN is stored in the private configuration, so it is not visible in the startup or running configuration.

Note A hand-generated startup configuration can contain the automatic login command for deployment purposes, but the **copy system:running-config nvram: startup-config** command must be issued to put the hand-generated configuration in the private configuration.

Manual Login

Manual login can be used when storing a PIN on the router is not desirable. Manual login can be executed with or without privileges, and it will make files and RSA keys on the eToken available to the Cisco IOS software. If a secondary configuration file is configured, it will only be executed with the privileges of the user who is performing the login. Thus, if you want to use manual login and set up the secondary configuration on the eToken to perform anything useful, you need to enable privileges.

Manual login can also be used in recovery scenarios for which the router configuration has been lost. If the scenario contains a remote site that normally connects to the core network with a VPN, the loss of the configuration and RSA keys requires out-of-band services that the eToken can provide. The eToken can contain a boot configuration, a secondary configuration, or both, and RSA keys to authenticate the connection.

Manual login may also be suitable for some initial deployment or hardware replacement scenarios for which the router is obtained from the local supplier or drop-shipped to the remote site.

Unlike automatic login, manual login requires that the user know the actual token PIN. However, if the user also has physical access to the eToken, he or she can use Aladdin's Windows-based utilities to copy the RSA keys and secondary config files from the eToken.

SUMMARY STEPS

1. **enable**
2. **crypto pki token** *token-name* [**admin**] **login** [*pin*]

   or

   configure terminal
3. **crypto pki token** *token-name* **user-pin** [*pin*]
4. exit
5. **show usbtoken**[**0-9**]**:***filename*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`crypto pki token`** *`token-name`* `[`**`admin`**`]` **`login`** `[`*`pin`*`]` **Example:** `Router# crypto pki token usbtoken0 admin login 5678` or **`configure terminal`** **Example:** `Router# configure terminal`	Manually logs into the eToken. You must specify the **admin** keyword if later you want to change the user PIN. or Puts the router in global configuration mode, which allows you to configure automatic eToken login.
Step 3	**`crypto pki token`** *`token-name`* **`user-pin`** `[`*`pin`*`]` **Example:** `Router(config)# crypto pki token usbtoken0 user-pin 1234`	(Optional) Creates a PIN that automatically allows the router to log into the USB eToken at router startup. **Note** Do not issue this command if you have already set up manual login.
Step 4	**`exit`** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	**`show usbtoken`**`[`**`0-9`**`]`**`:`***`filename`* **Example:** `Router#`	(Optional) Verifies whether the USB eToken has been logged onto the router.

Setting Administrative Functions on the eToken

Use this task to change default settings, such as the user PIN and the maximum number of failed on the eToken.

SUMMARY STEPS

1. **enable**
2. **crypto pki token** *token-name* [**admin**] **change-pin** [*pin*]
3. **configure terminal**
4. **crypto pki token** {*token-name* | **default**} **removal timeout** [*minutes*]
5. **crypto pki token** {*token-name* | **default**} **max-retries** [*number*]
6. **exit**
7. **copy usbflash**[**0-9**]**:***filename destination-url*

8. **show usbtoken**[**0-9**]**:***filename*
9. **crypto pki token** *token-name* **logout**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`crypto pki token`** *`token-name`* `[`**`admin`**`]` **`change-pin`** `[`*`pin`*`]` **Example:** `Router# crypto pki token usbtoken0 admin change-pin`	(Optional) Changes the user PIN number on the USB eToken. • If the PIN is not changed, the default PIN—1234567890—will be used. **Note** After the PIN has been changed, you must reset the login failure count to zero (via the **crypto pki token max-retries** command). The maximum number of allowable login failures is set (by default) to 15.
Step 3	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**`crypto pki token`** `{`*`token-name`* `|` **`default`**`}` **`removal timeout`** `[`*`seconds`*`]` **Example:** `Router(config)# crypto pki token usbtoken0 removal timeout 60`	(Optional) Sets the time interval, in seconds, that the router will wait before removing the RSA keys that are stored in the eToken after the eToken has been removed from the router. **Note** If this command is not issued, all RSA keys and IPSec tunnels associated with the eToken are torn down immediately after the eToken is removed from the router.
Step 5	**`crypto pki token`** `{`*`token-name`* `|` **`default`**`}` **`max-retries`** `[`*`number`*`]` **Example:** `Router(config)# crypto pki token usbtoken0 max-retries 20`	(Optional) Sets the maximum number of consecutive failed login attempts allowed before access to the eToken is denied. • By default, the value is set at 15.
Step 6	**`exit`** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 7	`copy usbflash[`**`0-9`**`]`**`:`***`filename destination-url`* **Example:** `Router# copy usbflash0:`	Copies files from the router to the eToken. • *destination-url*—See the **copy** command page documentation for a list of supported options.

	Command or Action	Purpose
Step 8	**show usbtoken**[**0-9**]**:***filename* **Example:** `Router#`	(Optional) Displays information about the USB eToken. You can use this command to verify whether the USB eToken has been logged onto the router.
Step 9	`crypto pki token` *token-name* `logout` **Example:** `Router# crypto pki toke usbtoken0 logout`	Logs the router out of the USB eToken. **Note** If you want to save any data to the USB eToken, you must log back into the eToken.

Troubleshooting USB Flash Drives and eTokens

This section contains descriptions of the following Cisco IOS commands that can be used to help troubleshoot possible problems that may arise while using a USB Flash or a USB eToken:

- The show file systems Command
- The show usb device Command
- The show usb controllers Command
- The dir Command

The show file systems Command

Step 1 Use the **show file systems** command to determine whether the router recognizes that there is a USB module plugged into a USB port. The USB module should appear on the list of file systems. If the module does not appear on the list, it can indicate any of the following problems:

- A connection problem with the USB module
- The Cisco IOS image running on the router does not support a USB module
- A hardware problem with the USB module itself

Step 2 Use the **show file systems** command to determine if a USB Flash module is formatted properly. To be compatible with a Cisco router, a USB Flash module must be formatted in a FAT16 format. If that is not the case, the **show file systems** command will display an error indicating an incompatible file system.

Sample output from the **show file systems** command showing a USB Flash module and a USB eToken appear below. The USB module listing appears in the last line of the examples.

```
Router# show file systems

File Systems:

     Size(b)      Free(b)       Type  Flags  Prefixes
           -            -     opaque     rw   archive:
           -            -     opaque     rw   system:
           -            -     opaque     rw   null:
           -            -    network     rw   tftp:
*   129880064     69414912       disk     rw   flash:#
      491512       486395      nvram     rw   nvram:
           -            -     opaque     wo   syslog:
           -            -     opaque     rw   xmodem:
           -            -     opaque     rw   ymodem:
```

```
       -          -   network     rw   rcp:
       -          -   network     rw   pram:
       -          -   network     rw   ftp:
       -          -   network     rw   http:
       -          -   network     rw   scp:
       -          -   network     rw   https:
       -          -    opaque     ro   cns:
63158272   33037312  usbflash     rw   usbflash0:
   32768        858  usbtoken     rw   usbtoken1:
```

The show usb device Command

Step 1 Use the **show usb device** command to determine if a USB module is supported by Cisco. The sample output for both the USB Flash and the USB eToken that indicates whether or not the module is supported are highlighted in the sample outputs below.

The following sample output is for a USB Flash module:

```
Router# show usb device

Host Controller:1
Address:0x1
Device Configured:YES
Device Supported:YES
Description:DiskOnKey
Manufacturer:M-Sys
Version:2.0
Serial Number:0750D84030316868
Device Handle:0x1000000
USB Version Compliance:2.0
Class Code:0x0
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x8EC
Product ID:0x15
Max. Packet Size of Endpoint Zero:64
Number of Configurations:1
Speed:Full
Selected Configuration:1
Selected Interface:0

Configuration:
    Number:1
    Number of Interfaces:1
    Description:
    Attributes:None
    Max Power:140 mA

    Interface:
        Number:0
        Description:
        Class Code:8
        Subclass:6
        Protocol:80
        Number of Endpoints:2

        Endpoint:
            Number:1
            Transfer Type:BULK
            Transfer Direction:Device to Host
```

```
        Max Packet:64
        Interval:0

    Endpoint:
        Number:2
        Transfer Type:BULK
        Transfer Direction:Host to Device
        Max Packet:64
        Interval:0
```

The following sample output is for a supported USB eToken:

```
Router# show usb device

Host Controller:1
Address:0x11
Device Configured:YES
Device Supported:YES
Description:eToken Pro 4254
Manufacturer:AKS
Version:1.0
Serial Number:
Device Handle:0x1010000
USB Version Compliance:1.0
Class Code:0xFF
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x529
Product ID:0x514
Max. Packet Size of Endpoint Zero:8
Number of Configurations:1
Speed:Low
Selected Configuration:1
Selected Interface:0

Configuration:
    Number:1
    Number of Interfaces:1
    Description:
    Attributes:None
    Max Power:60 mA

    Interface:
        Number:0
        Description:
        Class Code:255
        Subclass:0
        Protocol:0
        Number of Endpoints:0
```

The show usb controllers Command

Step 1 Use the **show usb controllers** command to determine if there is a hardware problem with a USB Flash module. If the **show usb controllers** command displays an error, it indicates a hardware problem in the USB module.

You can also use the **show usb controllers** command to verify that copy operations onto a USB Flash module are occurring successfully. Issuing the **show usb controllers** command after performing a file copy should display successful data transfers.

Sample output for the **show usb controllers** command for a working USB Flash module appears below:

```
Router# show usb controllers

Name:1362HCD
Controller ID:1
Controller Specific Information:
    Revision:0x11
    Control:0x80
    Command Status:0x0
    Hardware Interrupt Status:0x24
    Hardware Interrupt Enable:0x80000040
    Hardware Interrupt Disable:0x80000040
    Frame Interval:0x27782EDF
    Frame Remaining:0x13C1
    Frame Number:0xDA4C
    LSThreshold:0x628
    RhDescriptorA:0x19000202
    RhDescriptorB:0x0
    RhStatus:0x0
    RhPort1Status:0x100103
    RhPort2Status:0x100303
    Hardware Configuration:0x3029
    DMA Configuration:0x0
    Transfer Counter:0x1
    Interrupt:0x9
    Interrupt Enable:0x196
    Chip ID:0x3630
    Buffer Status:0x0
    Direct Address Length:0x80A00
    ATL Buffer Size:0x600
    ATL Buffer Port:0x0
    ATL Block Size:0x100
    ATL PTD Skip Map:0xFFFFFFFF
    ATL PTD Last:0x20
    ATL Current Active PTD:0x0
    ATL Threshold Count:0x1
    ATL Threshold Timeout:0xFF

Int Level:1
Transfer Completion Codes:
         Success              :920             CRC             :0
         Bit Stuff            :0               Stall           :0
         No Response          :0               Overrun         :0
         Underrun             :0               Other           :0
         Buffer Overrun       :0               Buffer Underrun :0
Transfer Errors:
         Canceled Transfers   :2               Control Timeout :0
Transfer Failures:
         Interrupt Transfer   :0               Bulk Transfer   :0
         Isochronous Transfer :0               Control Transfer:0
Transfer Successes:
         Interrupt Transfer   :0               Bulk Transfer   :26
         Isochronous Transfer :0               Control Transfer:894

USBD Failures:
         Enumeration Failures :0               No Class Driver Found:0
         Power Budget Exceeded:0

USB MSCD SCSI Class Driver Counters:
         Good Status Failures :3               Command Fail    :0
         Good Status Timed out:0               Device not Found:0
         Device Never Opened  :0               Drive Init Fail :0
         Illegal App Handle   :0               Bad API Command :0
```

```
         Invalid Unit Number  :0               Invalid Argument:0
         Application Overflow :0               Device in use   :0
         Control Pipe Stall   :0               Malloc Error    :0
         Device Stalled       :0               Bad Command Code:0
         Device Detached      :0               Unknown Error   :0
         Invalid Logic Unit Num:0

USB Aladdin Token Driver Counters:
         Token Inserted       :1               Token Removed   :0
         Send Insert Msg Fail :0               Response Txns   :434
         Dev Entry Add Fail   :0               Request Txns    :434
         Dev Entry Remove Fail:0               Request Txn Fail:0
         Response Txn Fail    :0               Command Txn Fail:0
         Txn Invalid Dev Handle:0

USB Flash File System Counters:
         Flash Disconnected   :0               Flash Connected :1
         Flash Device Fail    :0               Flash Ok        :1
         Flash startstop Fail :0               Flash FS Fail   :0

USB Secure Token File System Counters:
         Token Inserted       :1               Token Detached  :0
         Token FS success     :1               Token FS Fail   :0
         Token Max Inserted   :0               Create Talker Failures:0
         Token Event          :0               Destroy Talker Failures:0
         Watched Boolean Create Failures:0
```

The dir Command

Step 1 Use the **dir** command with the **usbflash**[**0-9**]**:** or the **usbtoken**[**0-9**]**:** keyword to display all files, directories, and their permission strings on the USB Flash or USB eToken.

The following sample output displays directory information for the USB Flash:

```
Router# dir usbflash0:

Directory of usbflash0:/

    1  -rw-    30125020  Dec 22 2032 05:31:32 +00:00  c3825-entservicesk9-mz.123-14.T

63158272 bytes total (33033216 bytes free)
```

The following sample output displays directory information for the USB eToken:

```
Router# dir usbtoken1:

Directory of usbtoken1:/

    2  d---          64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---        4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---           0  Dec 22 2032 05:23:40 +00:00  1002
   10  d---         512  Dec 22 2032 05:23:42 +00:00  1003
   12  d---           0  Dec 22 2032 05:23:42 +00:00  5000
   13  d---           0  Dec 22 2032 05:23:42 +00:00  6000
   14  d---           0  Dec 22 2032 05:23:42 +00:00  7000
   15  ----         940  Jun 27 1992 12:50:42 +00:00  mystartup-config
   16  ----        1423  Jun 27 1992 12:51:14 +00:00  myrunning-config

32768 bytes total (858 bytes free)
```

The following sample output displays directory information for all devices the router is aware of:

```
Router# dir all-filesystems

Directory of archive:/

No files in directory

No space information available
Directory of system:/

    2  drwx           0                    <no date>  its
  115  dr-x           0                    <no date>  lib
  144  dr-x           0                    <no date>  memory
    1  -rw-        1906                    <no date>  running-config
  114  dr-x           0                    <no date>  vfiles

No space information available
Directory of flash:/

    1  -rw-    30125020  Dec 22 2032 03:06:04 +00:00  c3825-entservicesk9-mz.123-14.T

129880064 bytes total (99753984 bytes free)
Directory of nvram:/

  476  -rw-        1947                    <no date>  startup-config
  477  ----          46                    <no date>  private-config
  478  -rw-        1947                    <no date>  underlying-config
    1  -rw-           0                    <no date>  ifIndex-table
    2  ----           4                    <no date>  rf_cold_starts
    3  ----          14                    <no date>  persistent-data

491512 bytes total (486395 bytes free)
Directory of usbflash0:/

    1  -rw-    30125020  Dec 22 2032 05:31:32 +00:00  c3825-entservicesk9-mz.123-14.T

63158272 bytes total (33033216 bytes free)
Directory of usbtoken1:/

    2  d---          64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---        4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---           0  Dec 22 2032 05:23:40 +00:00  1002
   10  d---         512  Dec 22 2032 05:23:42 +00:00  1003
   12  d---           0  Dec 22 2032 05:23:42 +00:00  5000
   13  d---           0  Dec 22 2032 05:23:42 +00:00  6000
   14  d---           0  Dec 22 2032 05:23:42 +00:00  7000
   15  ----         940  Jun 27 1992 12:50:42 +00:00  mystartup-config
   16  ----        1423  Jun 27 1992 12:51:14 +00:00  myrunning-config

32768 bytes total (858 bytes free)
```

Configuration Examples for Secure Token Support

This section contains the following configuration example:

- Logging Into and Saving RSA Keys to eToken: Example, page 16

Logging Into and Saving RSA Keys to eToken: Example

The following configuration example shows to how log into the eToken, generate RSA keys, and store the RSA keys onto the eToken:

```
! Configure the router to automatically log into the eToken
configure terminal
 crypto pki token default user-pin 0 1234567890
! Generate RSA keys and enroll certificates with the CA.
crypto pki trustpoint IOSCA
 enrollment url http://10.23.2.2
 exit
crypto ca authenticate IOSCA
Certificate has the following attributes:
       Fingerprint MD5:23272BD4 37E3D9A4 236F7E1A F534444E
      Fingerprint SHA1:D1B4D9F8 D603249A 793B3CAF 8342E1FE 3934EB7A

% Do you accept this certificate? [yes/no]:yes
Trustpoint CA certificate accepted.
crypto pki enroll
crypto pki enroll IOSCA
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.

Password:
Re-enter password:

% The subject name in the certificate will include:c2851-27.cisco.com
% Include the router serial number in the subject name? [yes/no]:no
% Include an IP address in the subject name? [no]:no
Request certificate from CA? [yes/no]:yes
% Certificate request sent to Certificate Authority
% The 'show crypto ca certificate IOSCA verbose' command will show the fingerprint.


*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint MD5:E6DDAB1B
 0E30EFE6 54529D8A DA787DBA
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint SHA1:3B0F33B
7 57C02A10 3935042B C4B6CD3D 61039251
*Jan 13 06:47:21.021:%PKI-6-CERTRET:Certificate received from Certificate Authority
! Issue the write memory command, which will automatically save the RSA keys to the eToken
! instead of private NVRAM.
Router# write memory
Building configuration...
[OK]

*Jan 13 06:47:29.481:%CRYPTO-6-TOKENSTOREKEY:Key c2851-27.cisco.com stored on
Cryptographic Token eToken Successfully
```

The following sample output from the **show crypto key mypubkey rsa** command displays stored credentials after they are successfully load from the eToken. Credentials that are stored on the eToken are in the protected area. When storing the credentials on the eToken, the files are stored in a directory called /keystore. However, the key files are hidden from the CLI.

```
Router# show crypto key mypubkey rsa

% Key pair was generated at:06:37:26 UTC Jan 13 2005
Key name:c2851-27.cisco.com
 Usage:General Purpose Key
```

```
 Key is not exportable.
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E3C644 43AA7DDD
  732E0F4E 3CA0CDAB 387ABF05 EB8F22F2 2431F1AE 5D51FEE3 FCDEA934 7FBD3603
  7C977854 B8E999BF 7FC93021 7F46ABF8 A4BA2ED6 172D3D09 B5020301 0001
% Key pair was generated at:06:37:27 UTC Jan 13 2005
Key name:c2851-27.cisco.com.server
 Usage:Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00DD96AE 4BF912EB
  2C261922 4784EF98 2E70E837 774B3778 7F7AEB2D 87F5669B BF5DDFBC F0D521A5
  56AB8FDC 9911968E DE347FB0 A514A856 B30EAFF4 D1F453E1 003CFE65 0CCC6DC7
  21FBE3AC 2F8DEA16 126754BC 1433DEF9 53266D33 E7338C95 BB020301 0001
```

Additional References

The following sections provide references related to USB storage support.

Related Documents

Related Topic	Document Title
Connecting the USB modules to the router	*Cisco Access Router USB Flash Module and USB eToken Hardware Installation Guide*
eToken and USB Flash data sheet	*USB eToken and USB Flash Features Support*
File management (loading, copying, and rebooting files)	*The section "File Management*" in the *Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*, Release 12.3
Configuring digital certificate encryption	The chapter "Configuring Certification Authority Interoperability" in the *Cisco IOS Security Configuration Guide*, Release 12.3

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

New Commands

- **crypto pki token change-pin**
- **crypto pki token login**
- **crypto pki token logout**
- **crypto pki token max-retries**
- **crypto pki token removal timeout**
- **crypto pki token secondary config**
- **crypto pki token user-pin**
- **debug usb driver**
- **show usb driver**
- **show usb controllers**
- **show usb device**
- **show usb driver**
- **show usb port**
- **show usbtoken**
- **show usb tree**

Modified Commands

- **boot config**
- **copy**
- **delete**
- **dir**
- **format**

Configuring Basic File Transfer Services

Configuring Basic File Transfer Services

Last Updated: May 2, 2008

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This module describes how to configure a router as a Trivial File Transfer Protocol (TFTP) or Reverse Address Resolution Protocol (RARP) server, configure the router to forward extended BOOTP requests over asynchronous interfaces, and configure rcp, rsh, and FTP in Cisco IOS Release 12.2.

For a complete description of the file transfer function commands mentioned in this chapter, refer to the *Cisco IOS Configuration Fundamentals Command Reference*.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

Basic File Transfer Services Configuration Task List

To configure basic file transfer services, perform any of the tasks described in the following sections:

- Configuring a Router as a TFTP or RARP Server
- Configuring System BOOTP Parameters
- Configuring a Router to Use rsh and rcp
- Configuring a Router to Use FTP Connections

All tasks in this chapter are optional.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Configuring a Router as a TFTP or RARP Server

It is too costly and inefficient to have a machine that acts only as server on every network segment. However, when you do not have a server on every segment, your network operations can incur substantial time delays across network segments. You can configure a router to serve as a RARP or TFTP server to reduce costs and time delays in your network while allowing you to use your router for its regular functions.

Typically, a router that is configured as a TFTP or RARP server provides other routers with system image or router configuration files from its Flash memory. You can also configure the router to respond to other types of service requests, such as requests.

Configuring a Router as a TFTP Server

As a TFTP server host, the router responds to TFTP Read Request messages by sending a copy of the system image contained in ROM or one of the system images contained in Flash memory to the requesting host. The TFTP Read Request message must use one of the filenames that are specified in the configuration.

Note For the Cisco 7000 family, the filename used must represent a software image that is present in Flash memory. If no image resides in Flash memory, the client router will boot the server's ROM image as a default.

Flash memory can be used as a TFTP file server for other routers on the network. This feature allows you to boot a remote router with an image that resides in the Flash server memory.

Some Cisco devices allow you to specify one of the different Flash memory locations (**bootflash:**, **slot0:**, **slot1:**, **slavebootflash:**, **slaveslot0:**, or **slaveslot1:**) as the TFTP server.

In the description that follows, one Cisco 7000 router is referred to as the *Flash server*, and all other routers are referred to as *client routers*. Example configurations for the Flash server and client routers include commands as necessary.

TFTP Router Configuration Prerequisite Tasks

The server and client router must be able to reach each other before the TFTP function can be implemented. Verify this connection by testing the connection between the server and client router (in either direction) using the **ping** *a.b.c.d* command (where *a.b.c.d* is the address of the client device). After the **ping** command is issued, connectivity is indicated by a series of exclamation points (!), while a series of periods (.) plus `[timed out]` or `[failed]` indicates that the connection attempt failed. If the connection fails, reconfigure the interface, check the physical connection between the Flash server and client router, and ping again.

After you verify the connection, ensure that a TFTP-bootable image is present on the server. This is the system software image the client router will boot. Note the name of this software image so you can verify it after the first client boot.

Caution For full functionality, the software image sent to the client must be the same type as the ROM software installed on the client router. For example, if the server has X.25 software, and the client does not have X.25 software in ROM, the client will not have X.25 capabilities after booting from the server's image in Flash memory.

Enabling the TFTP Server

To enable TFTP server operation, use the following commands, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **tftp-server flash** [*partition-number***:**]*filename1* [**alias** *filename2*] [*access-list-number*] or Router(config)# **tftp-server flash** *device***:***filename* (Cisco 7000 family only) or Router(config)# **tftp-server flash** [*device***:**][*partition-number***:**]*filename* (Cisco 1600 series and Cisco 3600 series only) or Router(config)# **tftp-server rom alias** *filename1* [*access-list-number*]	Specifies the system image to send in response to Read Requests. You can enter multiple lines to specify multiple images.
Step 3	Router(config)# **end**	Ends the configuration session and returns you to privileged EXEC mode.
Step 4	Router# **copy running-config startup-config**	Saves the running configuration to the startup configuration file.

The TFTP session can sometimes fail. TFTP generates the following special characters to help you determine why a TFTP session fails:

- An “E” character indicates that the TFTP server received an erroneous packet.
- An “O” character indicates that the TFTP server received an out-of-sequence packet.
- A period (.) indicates a timeout.

For diagnosing any undue delay in the transfer, the output is useful. For troubleshooting procedures, refer to the *Internetwork Troubleshooting Guide* publication.

In the following example, the system can use TFTP to send copies of the Flash memory file *version-10.3* in response to a TFTP Read Request for that file. The requesting host is checked against access list 22.

```
tftp-server flash version-10.3 22
```

In the following example, the system can use TFTP to send a copy of the ROM image *gs3-k.101* in response to a TFTP Read Request for the *gs3-k.101* file:

```
tftp-server rom alias gs3-k.101
```

The following example a router to send a copy of the file *gs7-k.9.17* in Flash memory in response to a TFTP Read Request. The client router must reside on a network specified by access list 1. Thus, in the example, the any clients on network 172.16.101.0 are permitted access to the file.

```
Server# configure terminal
Enter configuration commands, one per line. End with CTRL/Z
Server(config)# tftp-server flash gs7-k.9.17 1
Server(config)# access-list 1 permit 172.16.101.0 0.0.0.255
```

```
Server(config)# end
Server# copy running-config startup-config
[ok]
Server#
```

Configuring the Client Router

Configure the client router to first load a system image from the server. As a backup, configure the client router to then load its own ROM image if the load from a server fails. To configure the client router, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# no boot system`	(Optional) Removes all previous **boot system** statements from the configuration file.
Step 3	`Router(config)# boot system [tftp] filename [ip-address]`	Specifies that the client router load a system image from the server.
Step 4	`Router(config)# boot system rom`	Specifies that the client router loads its own ROM image if the load from a server fails.
Step 5	`Router(config)# config-register value`	Sets the configuration register to enable the client router to load a system image from a network server.
Step 6	`Router(config)# end`	Exits global configuration mode.
Step 7	`Router# copy running-config startup-config`	Saves the configuration file to your startup configuration.
Step 8	`Router# reload`	(Optional) Reloads the router to make your changes take effect.

After the system reloads, you should use the **show version** EXEC mode command to verify that the system booted the desired image.

Caution Using the **no boot system** command, as in the following example, will invalidate *all* other boot system commands currently in the client router system configuration. Before proceeding, determine whether the system configuration stored in the client router should first be saved (uploaded) to a TFTP file server so you have a backup copy.

In the following example, the router is configured to boot from a specified TFTP server:

```
Client# configure terminal
Enter configuration commands, one per line. End with CTRL/Z
Client(config)# no boot system
Client(config)# boot system c5300-js-mz.121-5.T.bin 172.16.111.111
Client(config)# boot system rom
Client(config)# config-register 0x010F
Client(config)# end
Client# copy running-config startup-config
[ok]
Client# reload
```

In this example, the **no boot system** command invalidates all other **boot system** commands currently in the configuration memory, and any **boot system** commands entered after this command will be executed first. The second command, **boot system** *filename address*, tells the client router to look for the file `c5300-js-mz.121-5.T.bin` on the TFTP server with an IP address of 172.16.111.111. Failing this, the client router will boot from its system ROM in response to the **boot system rom** command, which is included as a backup in case of a network problem. The **copy running-config startup-config** command copies the configuration to the startup configuration, and the **reload** command boots the system.

Note The system software to be booted from the server must reside in Flash memory on the server. If it is not in Flash memory, the client router will boot the server's system ROM.

The following example shows sample output of the **show version** command after the router has rebooted:

```
Router> show version

Cisco Internetwork Operating System Software
Cisco IOS (tm) 5300 Software (C5300-JS-M), Version 12.1(5)T,  RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2000 by Cisco Systems, Inc.
Compiled Sat 11-Nov-00 03:03 by joe
Image text-base: 0x60008958, data-base: 0x611C6000

ROM: System Bootstrap, Version 11.2(9)XA, RELEASE SOFTWARE (fc2)
BOOTFLASH: 5300 Software (C5300-BOOT-M), Version 12.0(7)T,  RELEASE SOFTWARE (f)

Router uptime is 8 weeks, 4 days, 22 hours, 36 minutes
System returned to ROM by power-on
System restarted at 00:37:38 UTC Thu Feb 22 2001
System image file is "flash:c5300-js-mz.121-5.T.bin"

.
.
.

Configuration register is 0x010F
```

The important information in this example is contained in the first line "Cisco IOS (tm).." and in the line that begins "System image file...." The "Cisco IOS (tm)..." line shows the version of the operating system in NVRAM. The "System image file...." line show the filename of the system image loaded from the TFTP server.

Configuring a Router as a RARP Server

Reverse Address Resolution Protocol (RARP) is a protocol in the TCP/IP stack that provides a method for finding IP addresses based on MAC (physical) addresses. This functionality is the reverse of broadcasting Address Resolution Protocols (ARPs), through which a host can dynamically discover the MAC-layer address corresponding to a particular IP network-layer address. RARP makes diskless booting of various systems possible (for example, diskless workstations that do not know their IP addresses when they boot, such as Sun workstations or PCs on networks where the client and server are on separate subnets). RARP relies on the presence of a RARP server with cached table entries of MAC-layer-to-IP address mappings.

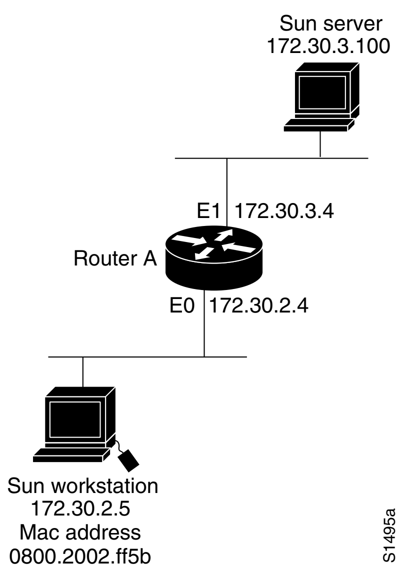
You can configure a Cisco router as a RARP server. This feature enables the Cisco IOS software to answer RARP requests.

To configure the router as a RARP server, use the following commands, beginning in global configuration mode:

Command	Purpose
`Router(config)# interface type [slot/]port`	Specifies the interface that you will be configuring the RARP service on and enters interface configuration mode for the specified interface.
`Router(config-if)# ip rarp-server ip-address`	Enables the RARP service on the router.

Figure 13 illustrates a network configuration in which a router is configured to act as a RARP server for a diskless workstation. In this example, the Sun workstation attempts to resolve its MAC (hardware) address to an IP address by sending a SLARP request, which is forwarded by the router to the Sun server.

Figure 13 Configuring a Router As a RARP Server



Router A has the following configuration:

```
! Allow the router to forward broadcast portmapper requests
ip forward-protocol udp 111
! Provide the router with the IP address of the diskless sun
arp 172.30.2.5 0800.2002.ff5b arpa
interface ethernet 0
! Configure the router to act as a RARP server, using the Sun Server's IP
! address in the RARP response packet.
ip rarp-server 172.30.3.100
! Portmapper broadcasts from this interface are sent to the Sun Server.
ip helper-address 172.30.3.100
```

The Sun client and server’s IP addresses must use the same major network number because of a limitation with the current SunOS *rpc.bootparamd* daemon.

In the following example, an access server is configured to act as a RARP server.

```
! Allow the access server to forward broadcast portmapper requests
 ip forward-protocol udp 111
! Provide the access server with the IP address of the diskless sun
 arp 172.30.2.5 0800.2002.ff5b arpa
 interface ethernet 0
! Configure the access server to act as a RARP server, using the Sun Server's
```

```
! IP address in the RARP response packet.
 ip rarp-server 172.30.3.100
! Portmapper broadcasts from this interface are sent to the Sun Server.
 ip helper-address 172.30.3.100
```

Configuring System BOOTP Parameters

The Boot Protocol (BOOTP) server for asynchronous interfaces supports extended BOOTP requests (defined in RFC 1084). The following command is useful in conjunction with using the auxiliary port as an asynchronous interface.

To configure extended BOOTP parameters for asynchronous interfaces, use the following command in global configuration mode:

Command	Purpose
Router(config)# **async-bootp** *tag* [:*hostname*] *data*	Configures extended BOOTP requests for asynchronous interfaces.

You can display the extended data that will be sent in BOOTP responses by using the following command in EXEC mode:

Command	Purpose
Router# **show async bootp**	Displays parameters for BOOTP responses.

For example, if the DNS server address is specified as extended data for BOOTP responses, you will see output similar to the following:

```
Router# show async bootp
The following extended data will be sent in BOOTP responses:

dns-server 172.22.53.210
```

For information about configuring your Cisco device as a BOOTP server, see the "Using AutoInstall and Setup" chapter.

Configuring a Router to Use rsh and rcp

Remote shell (rsh) gives users the ability to execute commands remotely. Remote copy (rcp) allows users to copy files to and from a file system residing on a remote host or server on the network. Cisco's implementation of rsh and rcp interoperates with the industry standard implementations. Cisco uses the abbreviation RCMD (Remote Command) to indicate both rsh and rcp.

This section is divided into the following sections:

- Specifying the Source Interface for Outgoing RCMD Communications
- About DNS Reverse Lookup for rcmd
- Enabling and Using rsh
- Enabling and Using rcp

Specifying the Source Interface for Outgoing RCMD Communications

You can specify the source interface for RCMD (rsh and rcp) communications. For example, the router can be configured so that RCMD connections use the loopback interface as the source address of all packets leaving the router. To specify the interface associated with RCMP communications, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`ip rcmd source-interface`** *`interface-id`*	Specifies the interface address that will be used to label all outgoing rsh and rcp traffic.

Specifying the source-interface is most commonly used to specify a loopback interface. This allows you to associate a permanent IP address with RCMD communications. Having a permanent IP address is useful for session identification (remote device can consistently idendify the origin of packets for the session). A "well-known" IP address can also be used for security purposes, as you can then create access lists on remote devices which include the address.

About DNS Reverse Lookup for rcmd

As a basic security check, the Cisco IOS software does a reverse lookup of the client IP address using DNS for the remote command (rcmd) applications (rsh and rcp). This check is performed using a host authentication process.

When enabled, the system records the address of the requesting client. That address is mapped to a host name using DNS. Then a DNS request is made for the IP address for that host name. The IP address received is then checked against the original requesting address. If the address does not match with any of the addresses received from DNS, the rcmd request will not be serviced.

This reverse lookup is intended to help protect against "spoofing." However, please note that the process only confirms that the IP address is a valid routable address; it is still possible for a hacker to spoof the valid IP address of a known host.

This feature is enabled by default. You can disable the DNS check for RCMD (rsh and rcp) access using the the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`no ip rcmd domain-lookup`**	Disables the Domain Name Service (DNS) reverse lookup function for remote command (rcmp) applications (rsh and rcp).

Enabling and Using rsh

You can use rsh (remote shell) to execute commands on remote systems to which you have access. When you issue the **rsh** command, a shell is started on the remote system. The shell allows you to execute commands on the remote system without having to log in to the target host.

You do not need to connect to the system, router, or access server and then disconnect after you execute a command if you use rsh. For example, you can use rsh to remotely look at the status of other devices *without* connecting to the target device, executing the command, and then disconnecting. This capability is useful for looking at statistics on many different routers. Configuration commands for enabling rsh use the acronym "rcmd", which is short for "remote command".

Maintaining rsh Security

To gain access to a remote system running rsh, such as a UNIX host, an entry must exist in the system's *.rhosts* file or its equivalent identifying you as a user who is authorized to execute commands remotely on the system. On UNIX systems, the *.rhosts* file identifies users who can remotely execute commands on the system.

You can enable rsh support on a router to allow users on remote systems to execute commands. However, our implementation of rsh does not support an *.rhosts* file. Instead, you must configure a local authentication database to control access to the router by users attempting to execute commands remotely using rsh. A local authentication database is similar to a UNIX *.rhosts* file. Each entry that you configure in the authentication database identifies the local user, the remote host, and the remote user.

Configuring the Router to Allow Remote Users to Execute Commands Using rsh

To configure the router as an rsh server, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`ip rcmd remote-host`** *`local-username`* {*`ip-address`* | *`host`*} *`remote-username`* [**`enable`** [*`level`*]]	Creates an entry in the local authentication database for each remote user who is allowed to execute rsh commands.
Step 2	`Router(config)#` **`ip rcmd rsh-enable`**	Enables the software to support incoming rsh commands.

To disable the software from supporting incoming rsh commands, use the **no ip rcmd rsh-enable** command.

Note When support of incoming rsh commands is disabled, you can still issue an rsh command to be executed on other routers that support the remote shell protocol and on UNIX hosts on the network.

The following example shows how to add two entries for remote users to the authentication database, and enable a router to support rsh commands from remote users:

```
ip rcmd remote-host Router1 172.16.101.101 rmtnetad1
ip rcmd remote-host Router1 172.16.101.101 netadmin4 enable
ip rcmd rsh-enable
```

The users, named *rmtnetad1* and *netadmin4*, are both on the remote host at IP address 172.16.101.101. Although both users are on the same remote host, you must include a unique entry for each user. Both users are allowed to connect to the router and remotely execute rsh commands on it after the router is enabled for rsh. The user named *netadmin4* is allowed to execute privileged EXEC mode commands on the router. Both authentication database entries give the router's host name *Router1* as the local username. The last command enables the router for to support rsh commands issued by remote users.

.

Executing Commands Remotely Using rsh

You can use rsh to execute commands remotely on network servers that support the remote shell protocol. To use this command, the *.rhosts* files (or equivalent files) on the network server must include an entry that permits you to remotely execute commands on that host.

If the remote server has a directory structure, as do UNIX systems, the rsh command that you issue is remotely executed from the directory of the account for the remote user that you specify through the **/user** *username* keyword and argument pair.

If you do not specify the **/user** keyword and argument, the Cisco IOS software sends a default remote username. As the default value of the remote username, the software sends the remote username associated with the current tty process, if that name is valid. If the tty remote username is invalid, the software uses the router host name as the both the remote and local usernames.

To execute a command remotely on a network server using rsh, use the following commands in user EXEC mode:

	Command	Purpose
Step 1	`Router> `**`enable`**` [`*`password`*`]`	Enters privileged EXEC mode.
Step 2	`Router# `**`rsh`**` {`*`ip-address`*` | `*`host`*`} [`**`/user`**` `*`username`*`] `*`remote-command`*	Executes a command remotely using rsh.

The following example executes the "ls -a" command in the home directory of the user sharon on mysys.cisco.com using rsh:

```
Router# enable
Router# rsh mysys.cisco.com /user sharon ls -a
.
..
.alias
.cshrc
.emacs
.exrc
.history
.login
.mailrc
.newsrc
.oldnewsrc
.rhosts
.twmrc
.xsession
jazz
Router#
```

Enabling and Using rcp

The remote copy (rcp) commands rely on the rsh server (or daemon) on the remote system. To copy files using rcp, you do not need to create a server for file distribution, as you do with TFTP. You need only to have access to a server that supports the remote shell (rsh). (Most UNIX systems support rsh.) Because you are copying a file from one place to another, you must have read permission on the source file and write permission in the destination directory. If the destination file does not exist, rcp creates it for you.

Although Cisco's rcp implementation emulates the functions of the UNIX rcp implementation—copying files among systems on the network—Cisco's command syntax differs from the UNIX rcp command syntax. The Cisco IOS software offers a set of copy commands that use rcp as the transport mechanism.

These rcp copy commands are similar in style to the Cisco IOS TFTP copy commands, but they offer an alternative that provides faster performance and reliable delivery of data. These improvements are possible because the rcp transport mechanism is built on and uses the Transmission Control Protocol/Internet Protocol (TCP/IP) stack, which is connection-oriented. You can use rcp commands to copy system images and configuration files from the router to a network server and vice versa.

You can also enable rcp support to allow users on remote systems to copy files to and from the router.

Configuring the Router to Accept rcp Requests from Remote Users

To configure the Cisco IOS software to support incoming rcp requests, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`ip rcmd remote-host`** *`local-username`* `{`*`ip-address`* `|` *`host`*`}` *`remote-username`* `[`**`enable`** `[`*`level`*`]]`	Create an entry in the local authentication database for each remote user who is allowed to execute rcp commands.
Step 2	`Router(config)#` **`ip rcmd rcp-enable`**	Enable the software to support incoming rcp requests.

To disable the software from supporting incoming rcp requests, use the **no ip rcmd rcp-enable** command.

Note When support for incoming rcp requests is disabled, you can still use the rcp commands to copy images from remote servers. The support for incoming rcp requests is distinct from its ability to handle outgoing rcp requests.

The following example shows how to add two entries for remote users to the authentication database and then enable the software to support remote copy requests from remote users. The users, named *netadmin1* on the remote host at IP address 172.16.15.55 and *netadmin3* on the remote host at IP address 172.16.101.101, are both allowed to connect to the router and remotely execute rcp commands on it after the router is enabled to support rcp. Both authentication database entries give the host name *Router1* as the local username. The last command enables the router to support for rcp requests from remote users.

```
ip rcmd remote-host Router1 172.16.15.55 netadmin1
ip rcmd remote-host Router1 172.16.101.101 netadmin3
ip rcmd rcp-enable
```

Configuring the Remote to Send rcp Requests

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy a configuration file from a server to the router using rcp, the Cisco IOS software sends the first valid username in the following list:

1. The username set by the **ip rcmd remote-username** command, if the command is configured.
2. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** command, the router software sends the Telnet username as the remote username.

Note In Cisco products, ttys are commonly used in access servers. The concept of tty originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called *tty devices*, which stands for *teletype*, the original UNIX terminal.

3. The router host name.

For **boot** commands using rcp, the software sends the router host name; you cannot explicitly configure the remote username.

For the rcp copy request to execute successfully, an account must be defined on the network server for the remote username.

If you are writing to the server, the rcp server must be properly configured to accept the rcp write request from the user on the router. For UNIX systems, you must add an entry to the *.rhosts* file for the remote user on the rcp server. For example, if the router contains the following configuration lines.

```
hostname Rtr1
ip rcmd remote-username User0
```

and the router's IP address translates to Router1.company.com, then the *.rhosts* file for User0 on the rcp server should contain the following line:

```
Router1.company.com Rtr1
```

Refer to the documentation for your rcp server for more details.

If the server has a directory structure, the configuration file or image is written or copied relative to the directory associated with the remote username on the server. Use the **ip rcmd remote-username** command to specify which directory on the server to use. For example, if the system image resides in the home directory of a user on the server, you can specify that user's name as the remote username.

If you copy the configuration file to a personal computer used as a file server, the computer must support rsh.

To override the default remote username sent on rcp requests, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`ip rcmd remote-username`** *`username`*	Specifies the remote username.

To remove the remote username and return to the default value, use the **no ip rcmd remote-username** command.

Configuring a Router to Use FTP Connections

You configure a router to transfer files between systems on the network using the File Transfer Protocol (FTP). With the Cisco IOS implementation of FTP, you can set the following FTP characteristics:

- Passive-mode FTP
- User name
- Password
- IP address

To configure these FTP characteristics, use any of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ip ftp username string`	Specifies the user name to be used for the FTP connection.
`Router(config)# ip ftp password [type] password`	Specifies the password to be used for the FTP connection.
`Router(config)# ip ftp passive` or `Router(config)# no ip ftp passive`	Configures the router to only use passive-mode FTP connections. or Allows all types of FTP connections (default).
`Router(config)# ip ftp source-interface interface`	Specifies the source IP address for FTP connections.

The following example demonstrates how to capture a core dump using the Cisco IOS FTP feature. The router accesses a server at IP address 192.168.10.3 with login name zorro and password sword. The default passive-mode FTP is used, and the server is accessed using Token Ring interface to1 on the router where the core dump will occur:

```
ip ftp username zorro
ip ftp password sword
ip ftp passive
ip ftp source-interface to1
! The following command allows the core-dump code to use FTP rather than TFTP or RCP
exception protocol ftp
! The following command creates the core dump in the event the system at IP address
! 192.168.10.3 crashes
exception dump 192.168.10.3
```

Transferring Files Using HTTP or HTTPS

Cisco IOS Release 12.4 provides the ability to transfer files between your Cisco IOS software-based device and a remote HTTP server using the HTTP or Secure HTTP (HTTPS) protocol. HTTP and HTTPS can now be specified as target or source locations in Cisco IOS command-line interface (CLI) commands that use file system prefixes such as the **copy** command.

Document Revision History

This document was first published on May 2, 2005, and last updated on May 2, 2005.

See the command reference documents for details on when support for specific commands was introduced. For details on when specific enhancements were integrated and where these enhancements appear in this document, see the "Feature Information for Transferring Files Using HTTP or HTTPS" section on page 12.

Contents

Prerequisites for Transferring Files Using HTTP or HTTPS

To copy files to or from a remote HTTP server, your system must support the HTTP client feature, which is integrated in most Cisco IOS software images. The HTTP client is enabled by default. To determine if the HTTP client is supported on your system, issue the **show ip http client all** command. If you are able to execute the command, the HTTP client is supported.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Commands exist for the optional configuration of the embedded HTTP client and for the HTTPS client, but the default configuration is sufficient for using the File Transfer Using HTTP or HTTPS feature. For information on configuring optional HTTP or HTTPS client characteristics, see the "Related Documents" section on page 11.

Restrictions for Transferring Files Using HTTP or HTTPS

Existing limitations to the **copy** command, such as no network-to-network copies, are in effect for the File Transfer Using HTTP or HTTPS feature.

Information About File Transfers Using HTTP or HTTPS

The File Transfer Using HTTP or HTTPS feature provides the capability to copy files, such as Cisco IOS image files, core files, configuration files, log files, scripts, and so on, to and from a remote server and your local routing device using the Cisco IOS **copy** command and command-line interface. The HTTP copy operation works in the same way as copying from other remote file systems, such as FTP or TFTP.

The HTTP copy operation can use the embedded HTTPS client for Secure HTTP transfers, providing secure and authenticated file transfers within the context of a public key infrastructure (PKI).

How to Transfer Files Using HTTP or HTTPS

To use the File Transfer Using HTTP feature, you may need to specify a username and password for the HTTP connections for those servers that require a username and password to connect. Commands are also available to specify custom connection characteristics, although default settings can be used. The feature also offers commands to monitor and maintain connections and files. These tasks are described in the following sections:

- Configuring HTTP Connection Characteristics for File Transfers, page 2 (as required)
- Downloading a File from a Remote Server Using HTTP or HTTPS, page 4 (required)
- Uploading a File to a Remote Server Using HTTP or HTTPS, page 6 (required)
- Maintaining and Monitoring File Transfers Using HTTP, page 8 (optional)

Configuring HTTP Connection Characteristics for File Transfers

In the following task, you will use configuration commands provided by the File Transfer Using HTTP or HTTPS feature to define connection characteristics. Default values are provided, but if you need to customize the connection characteristics for your network, the task in this section will help you specify a username and password, specify other connection characteristics such as connection preferences, configure a remote proxy server, and define the source interface to be used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http client connection** {**forceclose** | **idle timeout** *seconds* | **timeout** *seconds*}

4. **ip http client username** *username*
5. **ip http client password** *password*
6. **ip http client proxy-server** {*proxy-name* | *ip-address*} [**proxy-port** *port-number*]
7. **ip http client source-interface** *interface-id*
8. **do copy running-config startup-config**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip http client connection** {**forceclose** | **idle timeout** *seconds* | **timeout** *seconds*} **Example:** `Router(config)# ip http client connection timeout 15`	Configures characteristics for HTTP client connections to a remote HTTP server for all file transfers: • **forceclose**—Disables the default persistent connection. • **idle timeout** *seconds*—Sets the period of time allowed for an idle connection, in a range from 1 to 60 seconds. Default timeout is 30 seconds. • **timeout** *seconds*—Sets the maximum time the HTTP client will wait for a connection, in a range from 1 to 60 seconds. Default is 10 seconds.
Step 4	**ip http client username** *username* **Example:** `Router(config)# ip http client username user1`	Specifies the username to be used for HTTP client connections that require user authentication. **Note** You can also specify the username on the CLI when you issue the **copy** command, in which case the username entered overrides the username entered with this command. See the "Downloading a File with Username and Password in the CLI: Example" section on page 9 for an example.
Step 5	**ip http client password** *password* **Example:** `Router(config)# ip http client password letmein`	Specifies the password to be used for HTTP client connections that require user authentication. **Note** You can also specify the password on the CLI when you issue the **copy** command, in which case the password entered overrides the password entered with this command. See the "Downloading a File with Username and Password in the CLI: Example" section on page 9 for an example.

	Command or Action	Purpose
Step 6	**`ip http client proxy-server`** {*proxy-name* | *ip-address*} [**`proxy-port`** *port-number*] **Example:** `Router(config)# ip http client proxy-server edge2 proxy-port 29`	Configures the HTTP client to connect to a remote proxy server for HTTP file system client connections. • The optional **proxy-port** *port-number* keyword and argument specify the proxy port number on the remote proxy server.
Step 7	**`ip http client source-interface`** *interface-id* **Example:** `Router(config)# ip http client source-interface Ethernet 0/1`	Specifies the interface for the source address in all HTTP client connections.
Step 8	**`do copy running-config startup-config`** **Example:** `Router(config)# do copy running-config startup-config`	(Optional) Saves the running configuration as the startup configuration file. • The **do** command allows you to execute privileged EXEC mode commands from global configuration mode.
Step 9	**`end`** **Example:** `Router(config)# end` `Router#`	Ends your configuration session and returns the CLI to user EXEC mode.

Downloading a File from a Remote Server Using HTTP or HTTPS

This task downloads a file from a remote HTTP server using HTTP or HTTPS.

SUMMARY STEPS

1. **enable**
2. **copy** [**/erase**] [**/noverify**] **http://***remote-source-url local-destination-url*

   or

   copy https://*remote-source-url local-destination-url*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`copy`** `[`**`/erase`**`] [`**`/noverify`**`]` **`http://`***`remote-source-url`* *`local-destination-url`* or **`copy https://`***`remote-source-url`* *`local-destination-url`* **Example:** `Router# copy http://user1:mypassword@209.165.202.129:8080/image_files/c7200-i-mx flash:c7200-i-mx`	Copies a file from a remote web server to a local file system using HTTP or HTTPS. • **/erase**—Erases the local destination file system before copying. This option is provided on Class B file system platforms with limited memory to allow an easy way to clear local flash memory space. • **/noverify**—If the file being copied is an image file, this keyword disables the automatic image verification that occurs after an image is copied. • The *remote-source-url* argument is the location URL (or alias) from which to get the file to be copied, in standard Cisco IOS file system HTTP syntax as follows: **http://**[[*username*:*password*]@] {*hostname* | *host-ip*}[*/filepath*]*/filename* **Note** The optional *username* and *password* arguments can be used to log in to an HTTP server that requires user authentication, in place of configuring the **ip http client username** and **ip http client password** global configuration commands to specify these authentication strings. • The *local-destination-url* is the location URL (or alias) to put the copied file, in standard Cisco IOS file system syntax as follows: *filesystem*:[*/filepath*][*/filename*] **Note** For more information on URL syntax when you use the **copy** command, see the "Additional References" section on page 11.

Troubleshooting Tips

If file transfers from a remote web server fail, verify the following:

- Your router has an active connection to the Internet.
- The correct path and filename have been specified.
- The remote server requires a username and password.
- The remote server has a nonstandard communications port configured. (The default port for HTTP is 80; the default port for HTTPS is 443.)

The CLI will return error messages to help you determine the cause of a failed copy request. Additional information on the copy process can be displayed with the **debug ip http client all** command.

Uploading a File to a Remote Server Using HTTP or HTTPS

This task uploads a file to a remote HTTP server using HTTP or HTTPS.

SUMMARY STEPS

1. **enable**
2. **copy** [**/erase**] [**/noverify**] *local-source-url* **http://***remote-destination-url*

   or

   copy *local-source-url* **https://***remote-destination-url*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`copy`** `[`**`/erase`**`] [`**`/noverify`**`]` *`local-source-url`* **`http://`***`remote-destination-url`* or **`copy`** *`local-source-url`* **`https://`***`remote-destination-url`* **Example:** `Router# copy flash:c7200-i-mx http://user1:mypassword@209.165.202.129:8080/image_files/c7200-i-mx_backup`	Copies a file from a local file system to a remote web server using HTTP or HTTPS. • **/erase**—Erases the local destination file system before copying. This option is provided on Class B file system platforms with limited memory to allow an easy way to clear local flash memory space. • **/noverify**—If the file being copied is an image file, this keyword disables the automatic image verification that occurs after an image is copied. • The *local-source-url* argument is the location URL (or alias) from which to get the file to be copied, in standard Cisco IOS file system syntax as follows: **http://**[[*username*:*password*]@] {*hostname* | *host-ip*}[**/***filepath*]**/***filename* **Note** The optional *username* and *password* arguments can be used to log in to an HTTP server that requires user authentication, in place of configuring the **ip http client username** and **ip http client password** global configuration commands to specify these authentication strings. • The *remote-destination-url* is the URL (or alias) to put the copied file, in standard Cisco IOS file system syntax, as follows: *filesystem***:**[**/***filepath*][**/***filename*] **Note** For more information on URL syntax when you use the **copy** command, see the "Additional References" section on page 11.

Troubleshooting Tips

If file transfers from a remote web server fail, verify the following:

- Your router has an active connection to the Internet.
- The correct path and filename have been specified.
- The remote server requires a username and password.
- The remote server has a nonstandard communications port configured. (The default port for HTTP is 80; the default port for HTTPS is 443.)

The CLI will return error messages to help you determine the cause of a failed copy request. Additional information on the copy process can be displayed with the **debug ip http client all** command.

Maintaining and Monitoring File Transfers Using HTTP

Perform this task to maintain and monitor HTTP connections. Steps 2 through 4 can be performed in any order.

SUMMARY STEPS

1. **enable**
2. **show ip http client connection**
3. **show ip http client history**
4. **show ip http client session-module**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`show ip http client connection`** **Example:** `Router# show ip http client connection`	Displays details about active HTTP client connections.
Step 3	**`show ip http client history`** **Example:** `Router# show ip http client history`	Displays the last 20 URLs accessed by the HTTP client.
Step 4	**`show ip http client session-module`** **Example:** `Router# show ip http client session-module`	Displays details about about sessions (applications) that have registered with the HTTP client.

Configuration Examples for the File Transfer Using HTTP or HTTPS

This section provides the following configuration examples:

- Configuring HTTP Connection Characteristics: Example, page 9
- Downloading a File with Username and Password in the CLI: Example, page 9
- Downloading a File Using HTTP: Example, page 9
- Uploading a File Using HTTP: Example, page 9

Configuring HTTP Connection Characteristics: Example

In the following example, the HTTP password and username are configured for connection to a remote server that authenticates all users. The example also configures the connection for a 20-second idle connection period. The maximum time the HTTP client will wait for a connection remains at the default 10 seconds.

```
Router(config)# ip http client connection idle timeout 20
Router(config)# ip http client password Secret
Router(config)# ip http client username User1
Router(config)# do show running-config | include ip http client
```

Downloading a File with Username and Password in the CLI: Example

In the following example, the file c7200-i-mx is copied from a remote server to flash memory using HTTP. This example also shows how to enter a username and password from the command line for an HTTP server that authenticates users.

```
Router# copy http://user1:mypassword@209.165.202.129:8080/image_files/c7200-i-mx
flash:c7200-i-mx
```

Downloading a File Using HTTP: Example

The following example copies a file from the remote HTTP server to the flash memory. The example shows the prompts and displays that can be expected from transferring a file using the **copy** privileged EXEC command.

```
Router# copy http://172.19.209.190/user1/c7200-i-mz.test  flash:c7200-i-mz.test

Destination filename [c7200-i-mz.test]?
Loading http://172.19.209.190/user1/c7200-i-mz.test
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!
.
.
.
11272788 bytes copied in 527.104 secs (21386 bytes/sec)
```

Uploading a File Using HTTP: Example

The following example copies a file from flash memory to the remote HTTP server. The example shows the prompts and displays that can be expected from transferring a file using the **copy** privileged EXEC command.

```
Router# copy flash:c7200-js-mz.ELL2 http://172.19.209.190/user1/c7200-js-mz.ELL2
```

```
Address or name of remote host [172.19.209.190]?
Destination filename [user1/c7200-js-mz.ELL2]?
Storing http://172.19.209.190/user1/c7200-js-mz.ELL2 !!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
17571956 bytes copied in 57.144 secs (307503 bytes/sec)
```

Additional References

The following sections provide information related to transferring files using HTTP or HTTPS.

Related Documents

Related Topic	Document Title
Secure HTTP communications	*HTTPS – HTTP Server and Client with SSL 3.0*, Release 12.2(15)T feature document
Cisco IOS embedded web server	*HTTP 1.1 Web Server and Client*, Release 12.2(15)T feature document
Cisco IOS embedded web client	*HTTP 1.1 Client*

Standards

Standards	Title
No new of modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No relevant MIBs	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2616	*Hypertext Transfer Protocol -- HTTP/1.1*, R. Fielding, et al.
RFC 2617	*HTTP Authentication: Basic and Digest Access Authentication*, J. Franks, et al.

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Transferring Files Using HTTP or HTTPS

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or later appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific commands was introduced, see the command reference documents.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Transferring Files Using HTTP or HTTPS

Feature Name	Releases	Feature Configuration Information
HTTP/1.1 Client feature	12.2(15)T	The following sections provide information about this feature: "Configuring HTTP Connection Characteristics for File Transfers" section on page 2
HTTP Server and Client with SSL 3.0 (HTTPS) feature	12.2(15)T	The following sections provide information about this feature: "Configuring HTTP Connection Characteristics for File Transfers" section on page 2
File Download Using HTTP feature	12.3(2)T	This feature provides that files can be copied from an HTTP server to a Cisco IOS software-based platform. The following sections provide information about this feature: • "Downloading a File Using HTTP: Example" section on page 9

Table 1 Feature Information for Transferring Files Using HTTP or HTTPS (continued)

Feature Name	Releases	Feature Configuration Information
File Upload Using HTTP feature	12.3(7)T	The following sections provide information about this feature: • "Uploading a File to a Remote Server Using HTTP or HTTPS" section on page 6
File Transfer Using HTTP	12.3(7)T	The File Transfer Using HTTP feature provides the capability to copy files, such as Cisco IOS image files, core files, configuration files, log files, scripts, and so on, to and from a remote server and your local routing device using the Cisco IOS **copy** command and command-line interface. The HTTP copy operation works in the same way as copying from other remote file systems, such as FTP or TFTP. This feature provides support for copying files from a Cisco IOS software-based platform to an HTTP server, using either HTTP or HTTPS. The following sections provide information about this feature: • "Information About File Transfers Using HTTP or HTTPS" section on page 2 • "How to Transfer Files Using HTTP or HTTPS" section on page 2

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **copy http://**
- **copy https://**
- **debug ip http client**
- **ip http client connection**
- **ip http client password**
- **ip http client proxy-server**
- **ip http client source-interface**
- **ip http client username**
- **show ip http client connection**
- **show ip http client history**
- **show ip http client session-module**

ACL Authentication of Incoming rsh and rcp Requests

Feature History

Release	Modification
12.2(8)T	This feature was introduced.

This document describes the ACL Authentication of Incoming RSH and RCP Requests feature in Cisco IOS Release 12.2(8)T. It includes the following sections:

Feature Overview

To enable the Cisco IOS software to receive incoming remote shell (rsh) protocol and remote copy (rcp) protocol requests, customers must configure an authentication database to control access to the router. This configuration is accomplished by using the **ip rcmd remote-host** command.

Currently, when using this command, customers must specify the local user, the remote host, and the remote user in the database authentication configuration. For users who can execute commands to the router from multiple hosts, multiple database authentication configuration entries must be used, one for each host, as shown below.

```
ip rcmd remote-host local-user1 remote-host1 remote-user1
ip rcmd remote-host local-user1 remote-host2 remote-user1
ip rcmd remote-host local-user1 remote-host3 remote-user1
ip rcmd remote-host local-user1 remote-host4 remote-user1
```

This feature allows customers to specify an access list for a given user. The access list identifies the hosts to which the user has access. A new argument, *access-list*, has been added that can be used with this command to specify the access list, as shown below.

```
ip rcmd remote-host local-user1 access-list remote-user1
```


Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

To allow a user access to the hosts identified in the access list, first define the access list. If the access list is not already defined, access to the host will be denied. For information about defining an access list, refer to the *Cisco IOS Security Configuration Guide*, Release 12.2.

For more information about using the modified **ip rcmd remote-host** command, see the "Command Reference" section later in this document.

Related Documents

- *Cisco IOS Configuration Fundamentals Command Reference,* Release 12.2
- *Cisco IOS Security Configuration Guide*, Release 12.2
- *Cisco IOS Security Command Reference,* Release 12.2

Supported Platforms

- Cisco 805
- Cisco 806
- Cisco 828
- Cisco 1400 series
- Cisco 1600 series
- Cisco 1710
- Cisco 1720
- Cisco 1721
- Cisco 1750
- Cisco 1751
- Cisco 2420
- Cisco 3620
- Cisco 3631
- Cisco 3640
- Cisco 3660
- Cisco 3725
- Cisco 3745
- Cisco 2500 series
- Cisco 2600 series
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series
- Cisco uBR7200 series
- Cisco Voice Gateway 200
- URM (Universal Route Module)

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **ip rcmd remote-host**

Managing Configuration Files

Managing Configuration Files

Last Updated: May 2, 2008

This chapter describes how to create, load, and maintain configuration files. Configuration files contain a set of user-configured commands that customize the functionality of your Cisco routing device.

The tasks in this chapter assume that you have at least a minimal configuration running on your system. You can create a basic configuration file using the **setup** command (see Using Setup Mode to Configure a Cisco Networking Device for details).

For a complete description of the configuration file management commands in this chapter, refer to the *Cisco IOS Configuration Fundamentals Command Reference*.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see About Cisco IOS Software Documentation.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Managing Configuration Files" section on page 29.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Understanding Configuration Files

Configuration files contain the Cisco IOS software commands used to customize the functionality of your Cisco routing device (router, access server, switch, and so on). Commands are parsed (translated and executed) by the Cisco IOS software when the system is booted (from the startup-config file) or when you enter commands at the CLI in a configuration mode.

Types of Configuration Files

Startup configuration files (startup-config) are used during system startup to configure the software. Running configuration files (running-config) contain the current configuration of the software. The two configuration files can be different. For example, you may want to change the configuration for a short time period rather than permanently. In this case, you would change the running configuration using the **configure terminal** EXEC command but not save the configuration using the **copy running-config startup-config** EXEC command.

To change the running configuration, use the **configure terminal** command, as described in the "Modifying the Configuration File at the CLI" section later in this chapter. As you use the Cisco IOS configuration modes, commands generally are executed immediately and are saved to the running configuration file either immediately after you enter them or when you exit a configuration mode.

To change the startup configuration file, you can either save the running configuration file to the startup configuration using the **copy running-config startup-config** EXEC command or copy a configuration file from a file server to the startup configuration (see the "Copying Configuration Files from a Network Server to the Router" section for more information).

Location of Configuration Files

Configuration files are stored in the following locations:

- The running configuration is stored in RAM.
- On all platforms except the Class A Flash file system platforms, the startup configuration is stored in nonvolatile random-access memory (NVRAM).

- On Class A Flash file system platforms, the startup configuration is stored in the location specified by the CONFIG_FILE environment variable (see the "Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems" section for more information). The CONFIG_FILE variable defaults to NVRAM and can be a file in the following file systems:
  - **nvram:** (NVRAM)
  - **bootflash:** (internal Flash memory)
  - **slot0:** (first PCMCIA slot)
  - **slot1:** (second PCMCIA slot)

Configuration File Management Task List

To understand the management of Cisco IOS software configuration files, perform the tasks described in the following sections:

- Displaying Configuration File Information
- Entering Configuration Mode and Selecting a Configuration Source
- Modifying the Configuration File at the CLI
- Copying Configuration Files from the Router to a Network Server
- Copying Configuration Files from a Network Server to the Router
- Maintaining Configuration Files Larger than NVRAM
- Controlling the Parser Cache
- Copying Configuration Files Between Different Locations
- Reexecuting the Configuration Commands in the Startup Configuration File
- Clearing Configuration Information
- Specifying the Startup Configuration File

Displaying Configuration File Information

To display information about configuration files, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show bootvar`	Lists the contents of the BOOT environment variable, the name of the configuration file pointed to by the CONFIG_FILE environment variable, and the contents of the BOOTLDR environment variable.
`Router# more file-url`	Displays the contents of a specified file.

Command	Purpose
`Router# show running-config`	Displays the contents of the running configuration file. (Command alias for the **more system:running-config** command.)
`Router# show startup-config`	Displays the contents of the startup configuration file. (Command alias for the **more nvram:startup-config** command.) On all platforms except the Class A Flash file system platforms, the default startup-config file usually is stored in NVRAM. On the Class A Flash file system platforms, the CONFIG_FILE environment variable points to the default startup-config file. The CONFIG_FILE variable defaults to NVRAM.

Entering Configuration Mode and Selecting a Configuration Source

To enter configuration mode on the router, enter the **configure** command at the privileged EXEC prompt. The Cisco IOS software responds with the following prompt asking you to specify the terminal, memory, or a file stored on a network server (network) as the source of configuration commands:

```
Configuring from terminal, memory, or network [terminal]?
```

Configuring from the terminal allows you to enter configuration commands at the command line, as described in the following section. Configuring from memory loads the startup configuration file. See the "Reexecuting the Configuration Commands in the Startup Configuration File" section for more information. Configuring from the network allows you to load and execute configuration commands over the network. See the "Copying Configuration Files from a Network Server to the Router" section for more information.

Modifying the Configuration File at the CLI

The Cisco IOS software accepts one configuration command per line. You can enter as many configuration commands as you want.

You can add comments to a configuration file describing the commands you have entered. Precede a comment with an exclamation point (!). Because comments are *not* stored in NVRAM or in the active copy of the configuration file, comments do not appear when you list the active configuration with the **show running-config or more system:running-config** EXEC command. Comments do not display when you list the startup configuration with the **show startup-config** or **more nvram:startup-config** EXEC mode command. Comments are stripped out of the configuration file when it is loaded onto the router. However, you can list the comments in configuration files stored on a File Transfer Protocol (FTP), remote copy protocol (rcp), or Trivial File Transfer Protocol (TFTP) server.

When you configure the software using the CLI, the software executes the commands as you enter them. To configure the software using the CLI, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2		Enter the necessary configuration commands. The Cisco IOS documentation set describes configuration commands organized by technology.
Step 3	`Router(config)# end` or `Router(config)# ^Z`	Ends the configuration session and exits to EXEC mode. **Note** When you press the Ctrl and Z keys simultaneously, ^Z is displayed to the screen.
Step 4	`Router# copy system:running-config nvram:startup-config`	Saves the running configuration file as the startup configuration file. You may also use the **copy running-config startup-config** command alias, but you should be aware that this command is less precise. On most platforms, this command saves the configuration to NVRAM. On the Class A Flash file system platforms, this step saves the configuration to the location specified by the CONFIG_FILE environment variable (the default CONFIG_FILE variable specifies that the file should be saved to NVRAM).

In the following example, the router prompt name of the router is configured. The comment line, indicated by the exclamation mark (!), does not execute any command.

In this example, the **hostname** command is used to change the router name from Router to new_name. By pressing Ctrl-Z (^Z) or entering the **end** command, the user quits configuration mode. The **copy system:running-config nvram:startup-config** command saves the current configuration to the startup configuration.

```
Router# configure terminal
Router(config)# !The following command provides the router host name.
Router(config)# hostname new_name
new_name(config)# end
new_name# copy system:running-config nvram:startup-config
```

When the startup configuration is NVRAM, it stores the current configuration information in text format as configuration commands, recording only nondefault settings. The memory is checksummed to guard against corrupted data.

Note Some specific commands might not get saved to NVRAM. You will need to enter these commands again if you reboot the machine. These commands are noted in the documentation. We recommend that you keep a list of these settings so that you can quickly reconfigure your router after rebooting.

Copying Configuration Files from the Router to a Network Server

You can copy configuration files from the router to a file server using FTP, rcp, or TFTP. For example, you might perform this task to back up a current configuration file to a server before changing its contents, thereby allowing you to later restore the original configuration file from the server.

To copy configuration files from a router to a server, perform the tasks described in the following sections:

- Copying a Configuration File from the Router to a TFTP Server

- Copying a Configuration File from the Router to an rcp Server
- Copying a Configuration File from the Router to an FTP Server

The protocol you use depends on which type of server you are using. The FTP and rcp transport mechanisms provide faster performance and more reliable delivery of data than TFTP because FTP and rcp use the TCP/IP stack, which is connection-oriented.

Copying a Configuration File from the Router to a TFTP Server

In some implementations of TFTP, you must create a dummy file on the TFTP server and give it read, write, and execute permissions before copying a file over it. Refer to your TFTP documentation for more information.

To copy configuration information on a TFTP network server, use the following commands in the EXEC mode, as needed:

Command	Purpose
`Router#` **`copy system:running-config tftp:`**`[[[`**`//`***`location`*`]`**`/`***`directory`*`]`**`/`***`filename`*`]`	Copies the running configuration file to a TFTP server.
`Router#` **`copy nvram:startup-config tftp:`**`[[[`**`//`***`location`*`]`**`/`***`directory`*`]`**`/`***`filename`*`]`	Copies the startup configuration file to a TFTP server.

After you have issued the **copy** command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

The following example copies a configuration file from a router to a TFTP server:

```
Tokyo# copy system:running-config tftp://172.16.2.155/tokyo-confg
Write file tokyo-confg on host 172.16.2.155? [confirm] y

Writing tokyo-confg!!! [OK]
```

Copying a Configuration File from the Router to an rcp Server

You can copy configuration file from the router to an rcp server.

One of the first attempts to use the network as a resource in the UNIX community resulted in the design and implementation of the remote shell protocol, which included the remote shell (rsh) and remote copy (rcp) functions. Rsh and rcp give users the ability to execute commands remotely and copy files to and from a file system residing on a remote host or server on the network. The Cisco implementation of rsh and rcp interoperates with standard implementations.

The rcp **copy** commands rely on the rsh server (or daemon) on the remote system. To copy files using rcp, you need not create a server for file distribution, as you do with TFTP. You need only to have access to a server that supports the remote shell (rsh). (Most UNIX systems support rsh.) Because you are copying a file from one place to another, you must have read permission on the source file and write permission on the destination file. If the destination file does not exist, rcp creates it for you.

Although the Cisco rcp implementation emulates the functions of the UNIX rcp implementation—copying files among systems on the network—the Cisco command syntax differs from the UNIX rcp command syntax. The Cisco rcp support offers a set of **copy** commands that use rcp as the transport mechanism. These rcp **copy** commands are similar in style to the Cisco TFTP **copy** commands, but they offer an alternative that provides faster performance and reliable delivery of data. These improvements are possible because the rcp transport mechanism is built on and uses the TCP/IP stack, which is connection-oriented. You can use rcp commands to copy system images and configuration files from the router to a network server and vice versa.

You also can enable rcp support to allow users on remote systems to copy files to and from the router.

To configure the Cisco IOS software to allow remote users to copy files to and from the router, use the **ip rcmd rcp-enable** global configuration command.

About the rcp Username

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy a configuration file from the router to a server using rcp, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip rcmd remote-username** global configuration command, if the command is configured.
3. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** command, the router software sends the Telnet username as the remote username.
4. The router host name.

For the rcp copy request to execute successfully, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the remote username on the server. For example, if the system image resides in the home directory of a user on the server, you can specify that user name as the remote username.

Use the **ip rcmd remote-username** command to specify a username for all copies. (Rcmd is a UNIX routine used at the super-user level to execute commands on a remote machine using an authentication scheme based on reserved port numbers. Rcmd stands for “remote command”). Include the username in the **copy** command if you want to specify a username for that copy operation only.

If you are writing to the server, the rcp server must be properly configured to accept the rcp write request from the user on the router. For UNIX systems, you must add an entry to the .rhosts file for the remote user on the rcp server. For example, suppose the router contains the following configuration lines:

```
hostname Rtr1
ip rcmd remote-username User0
```

If the router IP address translates to Router1.company.com, then the .rhosts file for User0 on the rcp server should contain the following line:

```
Router1.company.com Rtr1
```

Refer to the documentation for your rcp server for more information.

Copying a Configuration File from the Router to an rcp Server

To copy a startup configuration file or a running configuration file from the router to an rcp server, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	(Optional) Enters global configuration mode.
Step 2	`Router(config)# ip rcmd remote-username username`	(Optional) Changes the default remote username.
Step 3	`Router(config)# end`	(Optional) Exits global configuration mode.
Step 4	`Router# copy system:running-config rcp:[[[//[username@]location]/directory]/filename]` or `Router# copy nvram:startup-config rcp:[[[//[username@]location]/directory]/filename]`	Specifies that the router running configuration file be stored on an rcp server. or Specifies that the router startup configuration file be stored on an rcp server.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Storing a Running Configuration File on an rcp Server Example

The following example copies the running configuration file named rtr2-confg to the netadmin1 directory on the remote host with an IP address of 172.16.101.101:

```
Router# copy system:running-config rcp://netadmin1@172.16.101.101/Rtr2-confg
Write file rtr2-confg on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
Router#
```

Storing a Startup Configuration File on an rcp Server Example

The following example shows how to store a startup configuration file on a server by using rcp to copy the file:

```
Rtr2# configure terminal
Rtr2(config)# ip rcmd remote-username netadmin2
Rtr2(config)# end
Rtr2# copy nvram:startup-config rcp:
Remote host[]? 172.16.101.101
Name of configuration file to write [rtr2-confg]?
Write file rtr2-confg on host 172.16.101.101?[confirm]
![OK]
```

Copying a Configuration File from the Router to an FTP Server

You can copy a configuration file from the router to an FTP server.

Understanding the FTP Username and Password

The FTP protocol requires a client to send a remote username and password on each FTP request to a server. When you copy a configuration file from the router to a server using FTP, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.

2. The username set by the **ip ftp username** global configuration command, if the command is configured.
3. Anonymous.

The router sends the first valid password it encounters in the following sequence:

1. The password specified in the **copy** command, if a password is specified.
2. The password set by the **ip ftp password** command, if the command is configured.
3. The router forms a password *username@routername.domain*. The variable *username* is the username associated with the current session, *routername* is the configured host name, and *domain* is the domain of the router.

The username and password must be associated with an account on the FTP server. If you are writing to the server, the FTP server must be properly configured to accept the FTP write request from the user on the router.

If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user name as the remote username.

Refer to the documentation for your FTP server for more information.

Use the **ip ftp username** and **ip ftp password** global configuration commands to specify a username and password for all copies. Include the username in the **copy** EXEC command if you want to specify a username for that copy operation only.

Copying a Configuration File from the Router to the FTP Server

To copy a startup configuration file or a running configuration file from the router to an FTP server, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	(Optional) Enters configuration mode from the terminal. This step is required only if you want to override the default remote username or password (see Steps 2 and 3).
Step 2	`Router(config)#` **`ip ftp username`** *`username`*	(Optional) Specifies the default remote username.
Step 3	`Router(config)#` **`ip ftp password`** *`password`*	(Optional) Specifies the default password.
Step 4	`Router(config)#` **`end`**	(Optional) Exits global configuration mode. This step is required only if you override the default remote username or password (see Steps 2 and 3).
Step 5	`Router#` **`copy system:running-config ftp:`**`[[[`**`//`**`[`*`username`*`[`**`:`***`password`*`]`**`@`**`]`*`location`*`]`**`/`***`directory`*`]`**`/`***`filename`*`]` or `Router#` **`copy nvram:startup-config ftp:`**`[[[`**`//`**`[`*`username`*`[`**`:`***`password`*`]`**`@`**`]`*`location`*`]`**`/`***`directory`*`]`**`/`***`filename`*`]`	Copies the running configuration or startup configuration file to an FTP server.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Storing a Running Configuration File on an FTP Server Example

The following example copies the running configuration file named rtr2-confg to the netadmin1 directory on the remote host with an IP address of 172.16.101.101:

```
Router# copy system:running-config ftp://netadmin1:mypass@172.16.101.101/Rtr2-confg
Write file rtr2-confg on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
Router#
```

Storing a Startup Configuration File on an FTP Server Example

The following example shows how to store a startup configuration file on a server by using FTP to copy the file:

```
Rtr2# configure terminal
Rtr2(config)# ip ftp username netadmin2
Rtr2(config)# ip ftp password mypass
Rtr2(config)# end
Rtr2# copy nvram:startup-config ftp:
Remote host[]? 172.16.101.101
Name of configuration file to write [rtr2-confg]?
Write file rtr2-confg on host 172.16.101.101?[confirm]
![OK]
```

Copying Configuration Files from a Network Server to the Router

You can copy configuration files from a TFTP, rcp, or FTP server to the running configuration or startup configuration of the router. You may want to perform this function for one of the following reasons:

- To restore a backed-up configuration file.
- To use the configuration file for another router. For example, you may add another router to your network and want it to have a similar configuration to the original router. By copying the file to the new router, you can change the relevant parts rather than re-creating the whole file.
- To load the same configuration commands on to all the routers in your network so that all the routers have similar configurations.

The **copy** {**ftp:** | **rcp:** | **tftp:**} **system:running-config** EXEC command loads the configuration files into the router as if you were typing the commands in at the command line. The router does not erase the existing running configuration before adding the commands. If a command in the copied configuration file replaces a command in the existing configuration file, the existing command will be erased. For example, if the copied configuration file contains a different IP address in a particular command than the existing configuration, the IP address in the copied configuration will be used. However, some commands in the existing configuration may not be replaced or negated. In this case, the resulting configuration file will be a mixture of the existing configuration file and the copied configuration file, with the copied configuration file having precedence.

In order to restore a configuration file to an exact copy of a file stored on a server, you need to copy the configuration file directly to the startup configuration (using the **copy** {**ftp:** | **rcp:** | **tftp:**} **nvram:startup-config** command) and reload the router.

To copy configuration files from a server to a router, perform the tasks described in the following sections:

- Copying a Configuration File from a TFTP Server to the Router
- Copying a Configuration File from an rcp Server to the Router
- Copying a Configuration File from an FTP Server to the Router

The protocol you use depends on which type of server you are using. The FTP and rcp transport mechanisms provide faster performance and more reliable delivery of data than TFTP. These improvements are possible because the FTP and rcp transport mechanisms are built on and use the TCP/IP stack, which is connection-oriented.

Copying a Configuration File from a TFTP Server to the Router

To copy a configuration file from a TFTP server to the router, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# copy tftp:[[[//location]/directory]/filename] system:running-config`	Copies a configuration file from a TFTP server to the running configuration.
`Router# copy tftp:[[[//location]/directory]/filename] nvram:startup-config`	Copies a configuration file from a TFTP server to the startup configuration.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

In the following example, the software is configured from the file named tokyo-config at IP address 172.16.2.155:

```
Router1# copy tftp://172.16.2.155/tokyo-confg system:running-config
Configure using tokyo-confg from 172.16.2.155? [confirm] y
Booting tokyo-confg from 172.16.2.155:!!! [OK - 874/16000 bytes]
```

Copying a Configuration File from an rcp Server to the Router

You can copy configuration files from an rcp server to the router.

Understanding the rcp Username

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy a configuration file from the router to a server using rcp, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip rcmd remote-username** global configuration command, if the command is configured.

3. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** command, the router software sends the Telnet username as the remote username.
4. The router host name.

For the rcp copy request to execute, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the remote username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user name as the remote username.

Copying a Configuration File from the rcp Server to the Router

To copy a configuration file from an rcp server to the running configuration or startup configuration, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# `**`configure terminal`**	(Optional) Enters configuration mode from the terminal. This step is required only if you override the default remote username (see Step 2).
Step 2	`Router(config)# `**`ip rcmd remote-username`** *`username`*	(Optional) Specifies the remote username.
Step 3	`Router(config)# `**`end`**	(Optional) Exits global configuration mode. This step is required only if you override the default remote username (see Step 2).
Step 4	`Router# `**`copy rcp:`**`[[[`**`//`**`[`*`username`***`@`**`]`*`location`*`]`**`/`***`directory`*`]`**`/`***`filename`*`]` **`system:running-config`** or `Router# `**`copy rcp:`**`[[[`**`//`**`[`*`username`***`@`**`]`*`location`*`]`**`/`***`directory`*`]`**`/`***`filename`*`]` **`nvram:startup-config`**	Copies the configuration file from a rcp server to the running configuration or startup configuration.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copy rcp Running-Config Example

The following example copies a configuration file named host1-confg from the netadmin1 directory on the remote server with an IP address of 172.16.101.101, and loads and runs those commands on the router:

```
Router# copy rcp://netadmin1@172.16.101.101/host1-confg system:running-config
Configure using host1-confg from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-confg:![OK]
Router#
%SYS-5-CONFIG: Configured from host1-config by rcp from 172.16.101.101
```

Copy rcp Startup-Config Example

The following example specifies a remote username of netadmin1. Then it copies the configuration file named host2-confg from the netadmin1 directory on the remote server with an IP address of 172.16.101.101 to the startup configuration.

```
Rtr2# configure terminal
Rtr2(config)# ip rcmd remote-username netadmin1
Rtr2(config)# end
Rtr2# copy rcp: nvram:startup-config
Address of remote host [255.255.255.255]? 172.16.101.101
Name of configuration file[rtr2-confg]? host2-confg
Configure using host2-confg from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-confg:![OK]
[OK]
Rtr2#
%SYS-5-CONFIG_NV:Non-volatile store configured from host2-config by rcp from
172.16.101.101
```

Copying a Configuration File from an FTP Server to the Router

You can copy configuration files from an FTP server to the router.

Understanding the FTP Username and Password

The FTP protocol requires a client to send a remote username and password on each FTP request to a server. When you copy a configuration file from the router to a server using FTP, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip ftp username** global configuration command, if the command is configured.
3. Anonymous.

The router sends the first valid password it encounters in the following sequence:

1. The password specified in the **copy** EXEC command, if a password is specified.
2. The password set by the **ip ftp password** global configuration command, if the command is configured.
3. The router forms a password *username@routername.domain*. The variable *username* is the username associated with the current session, *routername* is the configured host name, and *domain* is the domain of the router.

The username and password must be associated with an account on the FTP server. If you are writing to the server, the FTP server must be properly configured to accept the FTP write request from the user on the router.

If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user name as the remote username.

Refer to the documentation for your FTP server for more information.

Use the **ip ftp username** and **ip ftp password** global configuration commands to specify a username and password for all copies. Include the username in the **copy** command if you want to specify a username for that copy operation only.

Copying a Configuration File from an FTP Server to the Router

To copy a configuration file from an FTP server to the running configuration or startup configuration, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	(Optional) Allows you to enter global configuration mode. This step is required only if you want to override the default remote username or password (see Steps 2 and 3).
Step 2	`Router(config)# ip ftp username username`	(Optional) Specifies the default remote username.
Step 3	`Router(config)# ip ftp password password`	(Optional) Specifies the default password.
Step 4	`Router(config)# end`	(Optional) Exits global configuration mode. This step is required only if you override the default remote username or password (see Steps 2 and 3).
Step 5	`Router# copy ftp:[[[//[username[:password]@]location]/directory]/filename] system:running-config` or `Router# copy ftp:[[[//[username[:password]@]location]/directory]/filename] nvram:startup-config`	Using FTP, copies the configuration file from a network server to running memory or the startup configuration.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copy FTP Running-Config Example

The following example copies a host configuration file named host1-confg from the netadmin1 directory on the remote server with an IP address of 172.16.101.101, and loads and runs those commands on the router:

```
Router# copy rcp://netadmin1:mypass@172.16.101.101/host1-confg system:running-config
Configure using host1-confg from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-confg:![OK]
Router#
%SYS-5-CONFIG: Configured from host1-config by ftp from 172.16.101.101
```

Copy FTP Startup-Config Example

The following example specifies a remote username of netadmin1. Then it copies the configuration file named host2-confg from the netadmin1 directory on the remote server with an IP address of 172.16.101.101 to the startup configuration.

```
Rtr2# configure terminal
Rtr2(config)# ip ftp username netadmin1
Rtr2(config)# ip ftp password mypass
Rtr2(config)# end
Rtr2# copy ftp: nvram:startup-config
Address of remote host [255.255.255.255]? 172.16.101.101
Name of configuration file[rtr2-confg]? host2-confg
```

```
Configure using host2-config from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-config:![OK]
[OK]
Rtr2#
%SYS-5-CONFIG_NV:Non-volatile store configured from host2-config by ftp from
172.16.101.101
```

Maintaining Configuration Files Larger than NVRAM

To maintain a configuration file that exceeds size of NVRAM, perform the tasks described in the following sections:

- Compressing the Configuration File
- Storing the Configuration in Flash Memory on Class A Flash File Systems
- Loading the Configuration Commands from the Network

Compressing the Configuration File

The **service compress-config** global configuration command specifies that the configuration file be stored compressed in NVRAM. Once the configuration file has been compressed, the router functions normally. When the system is booted, it recognizes that the configuration file is compressed, expands it, and proceeds normally. The **more nvram:startup-config** EXEC command expands the configuration before displaying it.

Before you compress configuration files, refer to the appropriate hardware installation and maintenance publication. Verify that your system's ROMs support file compression. If not, you can install new ROMs that support file compression.

To compress configuration files, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`service compress-config`**	Specifies that the configuration file be compressed.
Step 2	`Router(config)#` **`end`**	Exits global configuration mode.
Step 3	`Use FTP, rcp, or TFTP to copy the new configuration. If you try to load a configuration that is more than three times larger than the NVRAM size, the following error message is displayed:` `"[buffer overflow -` *`file-size/buffer-size`* `bytes]."` or `Router#` **`configure terminal`**	Enters the new configuration.
Step 4	`Router(config)#` **`copy system:running-config nvram:startup-config`**	When you have finished changing the running-configuration, saves the new configuration.

The size of the configuration must not exceed three times the NVRAM size. For a 128-KB size NVRAM, the largest expanded configuration file size is 384 KB.

The **service compress-config** global configuration command works only if you have Cisco IOS software Release 10 or later release boot ROMs. Installing new ROMs is a one-time operation and is necessary only if you do not already have Cisco IOS Release 10 in ROM. If the boot ROMs do not recognize a compressed configuration, the following message is displayed:

```
Boot ROMs do not support NVRAM compression Config NOT written to NVRAM
```

The following example compresses a 129-KB configuration file to 11 KB:

```
Router# configure terminal
Router(config)# service compress-config
Router(config)# end
Router# copy tftp://172.16.2.15/tokyo-confg system:running-config
Configure using tokyo-confg from 172.16.2.155? [confirm] y
Booting tokyo-confg from 172.16.2.155:!!! [OK - 874/16000 bytes]
Router# copy system:running-config nvram:startup-config
Building configuration...
Compressing configuration from 129648 bytes to 11077 bytes
[OK]
```

Storing the Configuration in Flash Memory on Class A Flash File Systems

On Class A Flash file system routers, you can store the startup configuration in Flash memory by setting the CONFIG_FILE environment variable to a file in internal Flash memory or Flash memory in a PCMCIA slot.

To store the startup configuration in Flash memory, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	Router# **copy nvram:startup-config** *flash-filesystem:filename*	Copies the current startup configuration to the new location to create the configuration file.
Step 2	Router# **configure terminal**	Enters global configuration mode.
Step 3	Router(config)# **boot config flash-filesystem:***filename*	Specifies that the startup configuration file be stored in Flash memory by setting the CONFIG_FILE variable.
Step 4	Router(config)# **end**	Exits global configuration mode.
Step 5	Use FTP, rcp, or TFTP to copy the new configuration. If you try to load a configuration that is more than three times larger than the NVRAM size, the following error message is displayed: "[buffer overflow - *file-size*/*buffer-size* bytes]." or Router# **configure terminal**	Enters the new configuration.
Step 6	Router# **copy system:running-config nvram:startup-config**	When you have finished changing the running-configuration, saves the new configuration.

See the "Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems" section for more information.

The following example stores the configuration file in slot 0:

```
Router# copy nvram:startup-config slot0:router-config
Router# configure terminal
Router(config)# boot config slot0:router-config
Router(config)# end
Router# copy system:running-config nvram:startup-config
```

Care must be taken when editing or changing a large configuration. Flash memory space is used every time a **copy system:running-config nvram:startup-config** EXEC command is issued. Because file management for Flash memory, such as optimizing free space, is not done automatically, you must pay close attention to available Flash memory. Use the **squeeze** command to reclaim used space. We recommend that you use a large-capacity Flash card of at least 20 MB.

Loading the Configuration Commands from the Network

You can also store large configurations on FTP, rcp, or TFTP servers and download them at system startup. To use a network server to store large configurations, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	Router# **copy system:running-config** {**ftp:** | **rcp:** | **tftp:**}	Saves the running configuration to an FTP, rcp, or TFTP server.
Step 2	Router# **configure terminal**	Enters global configuration mode.
Step 3	Router(config)# **boot network** {**ftp:**[[[**//**[*username*[**:***password*]**@**]*location*]**/***directory*]**/***filename*] | **rcp:**[[[**//**[*username***@**]*location*]**/***directory*]**/***filename*] | **tftp:**[[[**//***location*]**/***directory*]**/***filename*]}	Specifies that the startup configuration file be loaded from the network server at startup.
Step 4	Router(config)# **service config**	Enables the router to download configuration files at system startup.
Step 5	Router(config)# **end**	Exits global configuration mode.
Step 6	Router# **copy system:running-config nvram:startup-config**	Saves the configuration.

See the "Copying Configuration Files from the Router to a Network Server" and "Configuring the Router to Download Configuration Files" sections for more information on these commands.

Controlling the Parser Cache

The Cisco IOS command-line parser in the Cisco IOS software performs the translation and execution (parsing) of command lines. The Parser Cache feature was developed to rapidly process large configuration files, thereby dramatically improving load time.

The Parser Cache feature allows the rapid recognition and translation of configuration lines in a configuration file that differ slightly from previously used configuration lines (for example, pvc 0/100, pvc 0/101, and so on) by dynamically creating, caching, and reusing simplified parse graphs. This improvement is useful primarily for configuration files that repeat similar commands hundreds or thousands of times, such as cases in which thousands of virtual circuits must be configured for subinterfaces, or hundreds of access lists must be configured. Performance will improve the most for those files in which the same commands are used repeatedly but the numerical arguments change from command to command.

The Parser Cache is enabled by default on all platforms using Cisco IOS Release 12.1(5)T and later releases. However, users with Cisco devices that do not require large configuration files may want to disable the Parser Cache to free the resources used by this feature. (Memory used by this feature depends on the size of the configuration files parsed, but is generally less than 512 KB.)

To control the Parser Cache feature, perform the tasks described in the following sections. All of these tasks are optional:

- Clearing the Parser Cache
- Disabling the Parser Cache
- Reenabling the Parser Cache
- Monitoring the Parser

Clearing the Parser Cache

To free resources or to reset the parser cache memory, you may wish to clear the parse entries and hit/miss statistics stored by the Parser Cache feature. To clear the information stored by the Parser Cache feature, use the following command in privileged EXEC mode:

Command	Purpose
`Router#` **`clear parser cache`**	Clears the parse cache entries and hit/miss statistics stored for the Parser Cache feature.

Disabling the Parser Cache

The Parser Cache feature is enabled by default. To disable the Parser Cache feature, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`no parser cache`**	Disables the Parser Cache feature.

When the parser cache is disabled, the **no parser cache** command line is written to the running configuration file.

Tip If you wish to disable the parser cache to free system resources, you should clear the parser cache before issuing the **no parser cache** command. You will not be able to clear the parser cache after disabling it.

Reenabling the Parser Cache

To reenable the Parser Cache feature after disabling it, use the following command in global configuration mode:

Command	Purpose
`Router(config)# parser cache`	Enables the Parser Cache feature.

Monitoring the Parser

Statistics about the last configuration file parsed are kept in the system memory, along with hit/miss statistics on the commands parsed by the Parser Cache feature. "Hits" and "misses" refer to the matches that the parser cache was able to make to similar commands used previously in the configuration session. Those commands that are matched ("hits") be parsed more efficiently. The parser cache cannot improve the parse time for those commands it was unable to match ("misses").

To display the parser statistics, use the following command in privileged EXEC mode:

Command	Purpose
`Router# show parser statistics`	Displays statistics about the last configuration file parsed and the status of the Parser Cache feature.

The following example shows sample output from the **show parser statistics** command:

```
Router# show parser statistics
Last configuration file parsed:Number of Commands:1484, Time:1272 ms

Parser cache:disabled, 0 hits, 0 misses
```

The **show parser statistics** command displays two sets of data, as follows:

- The number of commands in the configuration file that was last copied into the running configuration, and the time it took for the system to parse them (a configuration file can be loaded into the running configuration at system startup, or by issuing commands such as the **copy** *source* **running-config** EXEC command).
- The status of the parser cache (enabled or disabled) and the number of command matches (hits or misses) since the system was started or since the parser cache was cleared.

In the example shown, the hit/miss statistics (0/0) do not match the number of commands in the last configuration file parsed (1484), which indicates that the last configuration file was loaded while the parser cache was disabled.

Copying Configuration Files Between Different Locations

On many platforms, you can copy configuration files from one Flash memory device, such as internal Flash memory or a Flash memory card in a PCMCIA slot, to other locations. You also can copy configuration files from an FTP, rcp, or TFTP server to Flash memory.

Copying Configuration Files from Flash Memory to the Startup or Running Configuration

To copy a configuration file from Flash memory directly to your startup configuration in NVRAM or your running configuration, enter one following commands in EXEC mode, as needed:

Command	Purpose
Router> **copy** *filesystem*:[*partition-number*:][*filename*] **nvram:startup-config**	Loads a configuration file directly into NVRAM.
Router> **copy** *filesystem*:[*partition-number*:][*filename*] **system:running-config**	Copies a configuration file to your running configuration.

The following example copies the file named ios-upgrade-1 from partition 4 of the Flash memory PC Card in slot 0 to the router startup configurations:

```
Router# copy slot0:4:ios-upgrade-1 nvram:startup-config

Copy 'ios-upgrade-1' from flash device
  as 'startup-config' ? [yes/no] yes
[OK]
```

Copying Configuration Files Between Flash Memory File Systems

On platforms with multiple Flash memory file systems, you can copy files from one Flash memory file system, such as internal Flash memory or a Flash memory card in a PCMCIA slot, to another Flash memory file system. Copying files to different Flash memory file systems lets you create backup copies of working configurations and duplicate configurations for other routers.

To copy a configuration file between Flash memory file systems, use the following commands in EXEC mode:

	Command	Purpose
Step 1	Router> **show** *source-filesystem*:	Displays the layout and contents of Flash memory to verify the filename.
Step 2	Router> **copy** *source-filesystem*:[*partition-number*:][*filename*] *dest-filesystem*:[*partition-number*:][*filename*]	Copies a configuration file between Flash memory devices.
Step 3	Router> **verify** *dest-filesystem*:[*partition-number*:][*filename*]	Verifies the checksum of the file you copied.

Note The source device and the destination device cannot be the same. For example, the **copy slot1: slot1:** command is invalid.

Copying a Configuration File Between Local Flash Memory Devices Example

The following example copies the file named running-config from partition 1 of internal Flash memory to partition 1 of slot 1 on a Cisco 3600 series router. In this example, the source partition is not specified, so the router prompts for the partition number.

```
Router# copy flash: slot1:

System flash

Partition   Size    Used      Free      Bank-Size  State          Copy Mode
  1         4096K   3070K     1025K     4096K      Read/Write     Direct
  2        16384K   1671K    14712K     8192K      Read/Write     Direct

[Type ?<no> for partition directory; ? for full directory; q to abort]
Which partition? [default = 1]

System flash directory, partition 1:
File  Length   Name/status
  1   3142748  dirt/network/mars-test/c3600-j-mz.latest
  2   850      running-config
[3143728 bytes used, 1050576 available, 4194304 total]

PCMCIA Slot1 flash directory:
File  Length   Name/status
  1   1711088  dirt/gate/c3600-i-mz
  2   850      running-config
[1712068 bytes used, 2482236 available, 4194304 total]

Source file name? running-config

Destination file name [running-config]?
Verifying checksum for 'running-config' (file # 2)...  OK

Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]

Copy 'running-config' from flash: device
  as 'running-config' into slot1: device WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
...erased
!
 [OK - 850/4194304 bytes]

Flash device copy took 00:00:30 [hh:mm:ss]
Verifying checksum...  OK (0x16)
```

Copying a Configuration File from a Server to Flash Memory Devices

To copy a configuration file from an FTP server to a Flash memory device, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	(Optional) Enters global configuration mode. This step is required only if you override the default remote username or password (see Steps 2 and 3).
Step 2	`Router(config)#` **`ip ftp username`** *`username`*	(Optional) Specifies the remote username.

	Command	Purpose
Step 3	`Router(config)#` **`ip ftp password`** *`password`*	(Optional) Specifies the remote password.
Step 4	`Router(config)#` **`end`**	(Optional) Exits configuration mode. This step is required only if you override the default remote username (see Steps 2 and 3).
Step 5	`Router#` **`copy ftp:`** `[[[`**`//`**`[`*`username`***`:`***`password`***`@`**`]`*`location`*`]`**`/`***`directory`*`]`**`/`***`filename`*`]` *`flash-filesystem`***`:`**`[`*`partition-number`***`:`**`][`*`filename`*`]`	Copies the configuration file from a network server to the Flash memory device using FTP.

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

To copy a configuration file from an rcp server to a Flash memory device, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	(Optional) Enters global configuration mode. This step is required only if you override the default remote username (see Step 2).
Step 2	`Router(config)#` **`ip rcmd remote-username`** *`username`*	(Optional) Specifies the remote username.
Step 3	`Router(config)#` **`end`**	(Optional) Exits global configuration mode. This step is required only if you override the default remote username (see Step 2).
Step 4	`Router#` **`copy rcp:`**`[[[`**`//`**`[`*`username`***`@`**`]`*`location`*`]`**`/`***`directory`*`]`**`/`***`filename`*`]` *`flash-filesystem`***`:`**`[`*`partition-number`***`:`**`][`*`filename`*`]`	Copies the configuration file from a network server to the Flash memory device using rcp. Reply to any router prompts for additional information or confirmation. The prompting will depending on how much information you provide in the **copy** command and the current setting of the **file prompt** command.

To copy a configuration file from a TFTP server to the router, use the following command in EXEC mode:

Command	Purpose
`Router>` **`copy tftp:`**`[[[`**`//`***`location`*`]`**`/`***`directory`*`]`**`/`***`filename`*`]` *`flash-filesystem`***`:`**`[`*`partition-number`***`:`**`][`*`filename`*`]`	Copies the file from a TFTP server to the Flash memory device. Reply to any router prompts for additional information or confirmation. The prompting will depending on how much information you provide in the **copy** command and the current setting of the **file prompt** command.

The following example shows the copying of the configuration file named router-config from a TFTP server to the Flash memory card inserted in slot 0 of the Network Processing Engine (NPE) or Route Switch Processor (RSP) card of a Cisco 7500 series router. The copied file is renamed new-config.

```
Router# copy tftp:router-config slot0:new-config
```

Reexecuting the Configuration Commands in the Startup Configuration File

To reexecute the commands located in the startup configuration file, use the following command in privileged EXEC m ode:

Command	Purpose
`Router#` **`configure memory`**	Reexecutes the configuration commands located in the startup configuration file.

Clearing Configuration Information

You can clear the configuration information from the startup configuration. If you reboot the router with no startup configuration, the router will enter the Setup command facility so that you can configure the router from scratch.

Clearing the Startup Configuration

To clear the contents of your startup configuration, use the following command in EXEC mode:

Command	Purpose
`Router>` **`erase nvram:`**	Clears the contents of your startup configuration.

For all platforms except the Class A Flash file system platforms, this command erases NVRAM. The startup configuration file cannot be restored once it has been deleted.

On Class A Flash file system platforms, when you use the **erase startup-config** EXEC command, the router erases or deletes the configuration pointed to by CONFIG_FILE environment variable. If this variable points to NVRAM, the router erases NVRAM. If the CONFIG_FILE environment variable specifies a Flash memory device and configuration filename, the router deletes the configuration file. That is, the router marks the file as “deleted,” rather than erasing it. This feature allows you to recover a deleted file.

Deleting a Specified Configuration File

To delete a specified configuration on a specific Flash device, use the following command in EXEC mode:

Command	Purpose
`Router>` **`delete`** *`flash-filesystem`***`:`***`filename`*	Deletes a specified configuration file on a specified Flash device.

On Class A and B Flash file systems, when you delete a specific file in Flash memory, the system marks the file as deleted, allowing you to later recover a deleted file using the **undelete** EXEC command. Erased files cannot be recovered. To permanently erase the configuration file, use the **squeeze** EXEC command.

On Class C Flash file systems, you cannot recover a file that has been deleted.

If you attempt to erase or delete the configuration file specified by the CONFIG_FILE environment variable, the system prompts you to confirm the deletion.

The following example deletes the file named myconfig from a Flash memory card inserted in slot 0:

```
Router# delete slot0:myconfig
```

Specifying the Startup Configuration File

Normally, the router uses the startup configuration file in NVRAM or the Flash file system specified by the CONFIG_FILE environment variable (Class A Flash file systems only) at startup. See the "Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems" section for more information on setting the CONFIG_FILE variable.

You can also configure the router to automatically request and receive two configuration files from the network server at startup. See the "Configuring the Router to Download Configuration Files" section for more information.

Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems

On Class A Flash file systems, you can configure the Cisco IOS software to load the startup configuration file specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM. To change the CONFIG_FILE environment variable, use the following commands beginning in EXEC mode:

	Command	Purpose
Step 1	Router> **copy** [*flash-url* | *ftp-url* | *rcp-url* | *tftp-url* | **system:running-config** | **nvram:startup-config**] *dest-flash-url*	Copies the configuration file to the Flash file system from which the router will load the file upon restart.
Step 2	Router# **configure terminal**	Enters global configuration mode.
Step 3	Router(config)# **boot config** *dest-flash-url*	Sets the CONFIG_FILE environment variable. This step modifies the runtime CONFIG_FILE environment variable.
Step 4	Router(config)# **end**	Exits global configuration mode.
Step 5	Router> **copy system:running-config nvram:startup-config**	Saves the configuration performed in Step 3 to the startup configuration.
Step 6	Router> **show bootvar**	(Optional) Allows you to verify the contents of the CONFIG_FILE environment variable.

After you specify a location for the startup configuration file, the **nvram:startup-config** command is aliased to the new location of the startup configuration file. The **more nvram:startup-config** EXEC command will display the startup configuration, regardless of its location. The **erase nvram:startup-config** EXEC command will erase the contents of NVRAM and delete the file pointed to by the CONFIG_FILE environment variable.

When you save the configuration using the **copy system:running-config nvram:startup-config** command, the router saves a complete version of the configuration file to the location specified by the CONFIG_FILE environment variable and a distilled version to NVRAM. A distilled version is one that does not contain access list information. If NVRAM contains a complete configuration file, the router prompts you to confirm your overwrite of the complete version with the distilled version. If NVRAM contains a distilled configuration, the router does not prompt you for confirmation and proceeds with overwriting the existing distilled configuration file in NVRAM.

Note If you specify a file in a Flash device as the CONFIG_FILE environment variable, every time you save your configuration file with the **copy system:running-config nvram:startup-config** command, the old configuration file is marked as "deleted," and the new configuration file is saved to that device. Eventually, Flash memory will be full, because the old configuration files still take up memory. Use the **squeeze** EXEC command to permanently delete the old configuration files and reclaim the space.

The following example copies the running configuration file to the first PCMCIA slot of the RSP card in a Cisco 7500 series router. This configuration is then used as the startup configuration when the system is restarted.

```
Router# copy system:running-config slot0:config2
Router# configure terminal
Router(config)# boot config slot0:config2
Router(config)# end
Router# copy system:running-config nvram:startup-config
[ok]
Router# show bootvar
BOOT variable = slot0:rsp-boot-m
CONFIG_FILE variable = nvram:
Current CONFIG_FILE variable = slot0:config2

Configuration register is 0x010F
```

Configuring the Router to Download Configuration Files

You can configure the router to load one or two configuration files at system startup. The configuration files are loaded into memory and read in as if you were typing the commands at the command line. Thus, the configuration for the router will be a mixture of the original startup configuration and the one or two downloaded configuration files.

Network Versus Host Configuration Files

For historical reasons, the first file the router downloads is called the network configuration file. The second file the router downloads is called the host configuration file. Two configuration files can be used when all of the routers on a network use many of the same commands. The network configuration file contains the standard commands used to configure all of the routers. The host configuration files contain the commands specific to one particular host. If you are loading two configuration files, the host configuration file should be the configuration file you want to have precedence over the other file. Both the network and host configuration files must reside on a network server reachable via TFTP, rcp, or FTP, and must be readable.

Understanding the rcp Username

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy a configuration file from the router to a server using rcp, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **boot network** or **boot host** global configuration command, if a username is specified.
2. The username set by the **ip rcmd remote-username** global configuration command, if the command is configured.
3. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** command, the router software sends the Telnet username as the remote username.
4. The router host name.

For the rcp copy request to execute, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the remote username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user name as the remote username.

If you copy the configuration file to a personal computer used as a file server, the computer must support rsh.

Understanding the FTP Username and Password

The FTP protocol requires a client to send a remote username and password on each FTP request to a server. When you copy a configuration file from the router to a server using FTP, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip ftp username** global configuration command, if the command is configured.
3. Anonymous.

The router sends the first valid password in the following list:

1. The password specified in the **copy** command, if a password is specified.
2. The password set by the **ip ftp password** command, if the command is configured.
3. The router forms a password *username@routername.domain*. The variable *username* is the username associated with the current session, *routername* is the configured host name, and *domain* is the domain of the router.

The username and password must be associated with an account on the FTP server. If you are writing to the server, the FTP server must be properly configured to accept the FTP write request from the user on the router.

If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user name as the remote username.

Refer to the documentation for your FTP server for more information.

Use the **ip ftp username** and **ip ftp password** commands to specify a username and password for all copies. Include the username in the **copy** command if you want to specify a username for that copy operation only.

Configuring the Router to Download Configuration Files

You can specify an ordered list of network configuration and host configuration filenames. The Cisco IOS software scans this list until it loads the appropriate network or host configuration file.

To configure the router to download configuration files at system startup, perform at least one of the tasks described in the following sections:

- Configuring the Router to Download the Network Configuration File
- Configuring the Router to Download the Host Configuration File

If the router fails to load a configuration file during startup, it tries again every 10 minutes (the default setting) until a host provides the requested files. With each failed attempt, the router displays the following message on the console terminal:

```
Booting host-confg... [timed out]
```

Refer to the *Internetwork Troubleshooting Guide* for troubleshooting procedures.

If there are any problems with the startup configuration file, or if the configuration register is set to ignore NVRAM, the router enters the Setup command facility. See the "Using the Setup Command Facility for Configuration Changes" chapter in this publication for details on the Setup command facility.

Configuring the Router to Download the Network Configuration File

To configure the Cisco IOS software to download a network configuration file from a server at startup, use the following commands in global configuration mode:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **boot network** {**ftp:**[[[**//**[*username*[**:***password*]**@**]*location*]**/***directory*]**/***filename*] | **rcp:**[[[**//**[*username***@**]*location*]**/***directory*]**/***filename*] | **tftp:**[[[**//***location*]**/***directory*]**/***filename*]}	Specifies the network configuration file to download at startup, and the protocol to be used (TFTP, rcp, or FTP).
Step 3	Router(config)# **service config**	Enables the system to automatically load the network file upon restart.
Step 4	Router(config)# **end**	Exits global configuration mode.
Step 5	Router# **copy system:running-config nvram:startup-config**	Saves the running configuration to the startup configuration file.

For Step 2, if you do not specify a network configuration filename, the Cisco IOS software uses the default filename network-confg. If you omit the address, the router uses the broadcast address.

You can specify more than one network configuration file. The software tries them in order entered until it loads one. This procedure can be useful for keeping files with different configuration information loaded on a network server.

Configuring the Router to Download the Host Configuration File

To configure the Cisco IOS software to download a host configuration file from a server at startup, use the following commands in global configuration mode:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **boot host** {**ftp:**[[[**//**[*username*[**:***password*]**@**]*location*]**/***directory*]**/***filename*] | **rcp:**[[[**//**[*username***@**]*location*]**/***directory*]**/***filename*] | **tftp:**[[[**//***location*]**/***directory*]**/***filename*] }	Specifies the host configuration file to download at startup, and the protocol to be used (FTP, rcp, or TFTP).
Step 3	Router(config)# **service config**	Enables the system to automatically load the host file upon restart.
Step 4	Router(config)# **end**	Exits global configuration mode.
Step 5	Router# **copy system:running-config nvram:startup-config**	Saves the running configuration to the startup configuration file.

If you do not specify a host configuration filename, the router uses its own name to form a host configuration filename by converting the name to all lowercase letters, removing all domain information, and appending "-confg." If no host name information is available, the software uses the default host configuration filename router-confg. If you omit the address, the router uses the broadcast address.

You can specify more than one host configuration file. The Cisco IOS software tries them in order entered until it loads one. This procedure can be useful for keeping files with different configuration information loaded on a network server.

Configuring the Router to Download Configuration Files at System Startup Example

In the following example, a router is configured to download the host configuration file named hostfile1 and the network configuration file named networkfile1. The router uses TFTP and the broadcast address to obtain the file.

```
Router# configure terminal
Router(config)# boot host tftp:hostfile1
Router(config)# boot network tftp:networkfile1
Router(config)# service config
Router(config)# end
Router# copy system:running-config nvram:startup-config
```

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password..	http://www.cisco.com/techsupport

Command Reference

For information about commands mentioned in this chapter, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

Feature Information for Managing Configuration Files

Table 1 lists the release history for features related to managing configuration files.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Configuration File Management Features

Feature Name	Releases	Feature Information
Parser Cache	Cisco IOS Cisco IOS XE Release 2.1	The Cisco IOS command-line parser in the Cisco IOS software performs the translation and execution (parsing) of command lines. The Parser Cache feature was developed to rapidly process large configuration files, thereby dramatically improving load time. For information about feature support in Cisco IOS software, use Feature Navigator. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following section provides information about this feature: • Controlling the Parser Cache

Configuration Generation Performance Enhancement

First Published: March 2004
Last Updated: May 2, 2008

The Configuration Generation Performance Enhancement feature assists configuration management by enabling faster collection of running configuration file information. This feature is especially useful in managing large networks with numerous interfaces configured.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuration Generation Performance Enhancement" section on page 7.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for Configuration Generation Performance Enhancement

The device on which the Configuration Generation Performance Enhancement feature is used must have enough memory available to store (cache) a large interface configuration file. For example, if the interface configurations take up 15 KB of memory, using this feature would require having an additional 15 KB of memory space available.

Information About Configuration Generation Performance Enhancement

Before enabling the Configuration Generation Performance Enhancement feature, you should understand the following concepts:

- Cisco IOS Software Configuration Storage, page 2
- Benefits of the Configuration Generation Performance Enhancement, page 2

Cisco IOS Software Configuration Storage

In the Cisco IOS software configuration model, the configuration state is maintained in a distributed manner, with each component storing its own configuration state. To retrieve configuration information, the software must poll every component to collect the distributed information. This configuration state retrieval operation is performed by a process known as nonvolatile generation (NVGEN), and it is used by command-line interface (CLI) commands such as **show running-configuration**, **write memory**, and **copy system:running-configuration** to display or copy the running system configuration. When invoked, NVGEN queries each system component and each instance of interface or other configuration objects. A running configuration file is constructed as NVGEN traverses the system performing these queries.

Benefits of the Configuration Generation Performance Enhancement

Before the Configuration Generation Performance Enhancement feature was introduced, NVGEN always had to query the entire system and could generate only a total configuration. The time required to process the running configuration creates performance problems for configuration management, because completion of the NVGEN operation can take many minutes.

The Configuration Generation Performance Enhancement feature reduces the execution time for NVGEN processes and is especially useful for managing large configuration files that contain numerous interface configurations. This feature provides faster execution of commands that process the running system configuration by caching interface configuration information in system memory, and by retrieving only configuration information that has changed.

How to Configure the Configuration Generation Performance Enhancement

This section contains the following procedure:

- Configuring the Configuration Generation Performance Enhancement, page 3 (required)

Configuring the Configuration Generation Performance Enhancement

Perform this task to enable the Configuration Generation Performance Enhancement.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parser config cache interface**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parser config cache interface** **Example:** `Router(config)# parser config cache interface`	Reduces the time required for the CLI to execute commands that manage the running system configuration, especially for large configuration files.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for the Configuration Generation Performance Enhancement

This section provides the following examples:

- Configuring the Configuration Generation Performance Enhancement: Example, page 4

- Verifying the Configuration Generation Performance Enhancement: Example, page 4

Configuring the Configuration Generation Performance Enhancement: Example

The following example shows how to enable the Configuration Generation Performance Enhancement feature:

```
Router(config)# parser config cache interface
```

Verifying the Configuration Generation Performance Enhancement: Example

You can verify that the **parser config cache interface** command has been enabled by checking for the command in the system configuration file displayed when you enter the **show running-configuration** EXEC command.

Note The first time you display the configuration file, you will not see much evidence of improvement in performance because the interface cache will be filled up. However, you will notice performance improvements when you enter subsequent NVGEN-type commands such as the **show running-configuration** EXEC command.

Each time the interface configuration of an changes, the cache of the specified interface is flushed. The other interface data remains cached as before. Entering an NVGEN-type command after modifying the interface configuration will once again not show much evidence of improvement until the next NVGEN-type command is entered.

```
Router# show running-configuration
!
!
parser config cache interface
!
!
```

Additional References

The following sections provide references related to Configuration Generation Performance Enhancement feature.

Related Documents

Related Topic	Document Title
System configuration file management	Managing Configuration Files
System configuration file management commands	The *Cisco IOS Configuration Fundamentals Command Reference* appropriate to your software release version.

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **parser config cache interface**

Feature Information for Configuration Generation Performance Enhancement

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for the Configuration Generation Performance Enhancement Feature

Feature Name	Releases	Feature Information
Configuration Generation Performance Enhancement	12.3(7)T 12.2(25)S 12.2(33)SRC 12.2(33)SB Cisco IOS XE Release 2.1	The Configuration Generation Performance Enhancement feature assists configuration management by enabling faster collection of running configuration file information. This feature is especially useful in managing large networks with numerous interfaces configured. In 12.2(33)SB, this feature was implemented on the Cisco 10000 series. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following sections provide information about this feature: • Information About Configuration Generation Performance Enhancement • How to Configure the Configuration Generation Performance Enhancement

Exclusive Configuration Change Access and Access Session Locking

First Published: February 28, 2005
Last Updated: May 2, 2008

Exclusive Configuration Change Access (also called the "Configuration Lock" feature) allows you to have exclusive change access to the Cisco IOS running configuration, preventing multiple users from making concurrent configuration changes.

The Access Session Locking addition to this feature extends the Exclusive Configuration Change Access feature such that **show** and **debug** commands entered by the user holding the configuration lock always have execution priority; **show** and **debug** commands entered by other users are only allowed to run after the processes initiated by the configuration lock owner have finished.

The Exclusive Configuration Change Access feature ("exposed lock") is complementary with the locking mechanism in the Configuration Replace and Configuration Rollback feature ("rollback lock").

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Exclusive Configuration Change Access and Access Session Locking" section on page 10.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About Exclusive Configuration Change Access and Access Session Locking

To use the Exclusive Configuration Change Access and Access Session Locking feature, you should understand the following concepts:

Exclusive Configuration Change Access Functionality

Devices running Cisco IOS software maintain a running configuration that determines the configuration state of the device. Changes to the running configuration alter the behavior of the device. Because Cisco IOS software allows multiple users to change the running configuration via the device CLI (including the device console and telnet SSH), in some operating environments it would be beneficial to prevent multiple users from making concurrent changes to the Cisco IOS running configuration. Temporarily limiting access to the Cisco IOS running configuration prevents inadvertent conflicts or cases where two users attempt to configure the same portion of the running configuration.

Exclusive configuration change access provides a mechanism to prevent concurrent configuration of Cisco IOS software by multiple users.

This feature provides exclusive change access to the Cisco IOS running configuration from the time you enter global configuration mode by using the **configure terminal** command. This gives the effect of a "configuration lock," preventing other users from changing the Cisco IOS running configuration. The configuration lock is automatically released when the user exits Cisco IOS configuration mode.

The Exclusive Configuration Change Access feature is enabled using the **configuration mode exclusive** command in global configuration mode. Exclusive Configuration Change Access can be set to **auto**, so that the Cisco IOS configuration mode is locked whenever anyone uses the **configure terminal** command, or it can be set to **manual**, so that the Cisco IOS configuration mode is locked only when the **configure terminal lock** command is issued.

The Exclusive Configuration Change Access feature is complementary with the locking mechanism for the Configuration Replace and Configuration Rollback feature introduced in Cisco IOS Release 12.2(25)S and 12.3(7)T.

Access Session Locking

Access Session Locking, in addition to preventing concurrent configuration access, provides an option to prevent simultaneous processes, such as a **show** command entered by another user, from executing while other configuration commands are being executed. When this feature is enabled, the commands entered by the user with the configuration lock (such as configuration commands) always have priority over commands entered by other users.

How to Use Exclusive Configuration Change Access and Access Session Locking

This section contains the following procedures:

- Enabling Exclusive Configuration Change Access and Access Session Locking, page 3 (required)
- Obtaining Exclusive Configuration Change Access, page 4 (optional)
- Monitoring and Troubleshooting the Exclusive Configuration Change Access and Access Session Locking Feature, page 5 (optional)

Enabling Exclusive Configuration Change Access and Access Session Locking

Perform this task to gain exclusive access to the Cisco IOS configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **configuration mode exclusive** {**auto** | **manual**}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**configuration mode exclusive** {**auto** | **manual**} **Example:** `Router(config)# configuration mode exclusive auto`	Enables exclusive configuration change access (configuration lock feature). When enabled, configuration sessions are performed in single-user (exclusive) mode. • The **auto** keyword automatically locks the configuration session whenever the **configure terminal** command is used. This is the default. • The **manual** keyword allows you to choose to lock the configuration session manually or leave it unlocked. If you use the **manual** keyword, you must perform the task described in the "Obtaining Exclusive Configuration Change Access" section on page 4.
Step 4	**end** **Example:** `Router(config)# end`	Ends your configuration session and returns the CLI to privileged EXEC mode.

Obtaining Exclusive Configuration Change Access

Perform this task to obtain exclusive configuration change access for the duration of your configuration session. Use of the **lock** keyword with the **configure terminal** command is only necessary if the exclusive configuration mode has been set to **manual** (see the "Enabling Exclusive Configuration Change Access and Access Session Locking" section).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **configure terminal lock**
4. Configure the system by entering your changes to the running configuration.
5. **end**
   or
   exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**configure terminal lock** **Example:** `Router(config)# configure terminal lock`	(Optional) Locks the Cisco IOS software in exclusive (single-user) mode. • This command can only be used if you have previously enabled configuration locking by using the **configuration mode exclusive** command. • Available only in Cisco IOS Release 12.3(14)T or later.
Step 4	Configure the system by entering your changes to the running configuration.	—
Step 5	**end** or **exit** **Example:** `Router(config)# end` `Router#` or **Example:** `Router(config)# exit` `Router#`	Ends your configuration session, automatically releases the session lock obtained in Step 1, and exits to privileged EXEC mode. **Note** Either the **end** command, the **exit** command, or the Ctrl-Z key combination releases the configuration lock. Use of the **end** command is recommended.

Monitoring and Troubleshooting the Exclusive Configuration Change Access and Access Session Locking Feature

Perform one or both of the steps in this task to monitor or troubleshoot the Exclusive Configuration Change Access and Access Session Locking feature.

SUMMARY STEPS

1. **show configuration lock**
2. **debug configuration lock**

DETAILED STEPS

Step 1 **show configuration lock**

Use this command to display the status and details of any current configuration locks, including the owner, user, terminal, lock state, and lock class.

If you cannot enter global configuration mode, you can use this command to determine if the configuration session is currently locked by another user, and who that user is.

```
Router# show configuration lock

Parser Configure Lock
-----------------------------------------------------
Owner PID                          : 3
User                               : unknown
TTY                                : 0
```

```
Type                              : EXCLUSIVE
State                             : LOCKED
Class                             : EXPOSED
Count                             : 1
Pending Requests                  : 0
User debug info                   : configure terminal
Session idle state                : TRUE
No of exec cmds getting executed  : 0
No of exec cmds blocked           : 0
Config wait for show completion   : FALSE
Remote ip address                 : Unknown
Lock active time (in Sec)         : 6
Lock Expiration timer (in Sec)    : 593
Router(config)#
```

Step 2 **debug configuration lock**

Use this command to enable debugging of Cisco IOS configuration locks (exposed class locks or rollback class locks).

```
Router# debug configuration lock

Session1 from console
==========================

Router# configure terminal lock
Configuration mode locked exclusively. The lock will be cleared once you exit out of
configuration mode using end/exit
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Parser : LOCK REQUEST in EXCLUSIVE mode
Parser: <configure terminal lock> - Config. Lock requested by process <3> client <PARSER
Client>
Parser: <configure terminal lock> - Config. Lock acquired successfully !
Router(config)#
```

Configuration Examples for Exclusive Configuration Change Access and Access Session Locking

This section provides the following configuration examples:

- Configuring an Exclusive Lock in Auto Mode: Example, page 6
- Configuring an Exclusive Lock in Manual Mode: Example, page 7

Configuring an Exclusive Lock in Auto Mode: Example

The following example shows how to enable the exclusive lock in auto mode for single-user auto configuration mode using the **configuration mode exclusive auto** command. Once the Cisco IOS configuration file is locked exclusively, you can verify this configuration by using the **show configuration lock** command.

```
Router#
Router# configure terminal
Router(config)# configuration mode exclusive auto
Router(config)# exit
```

```
Router#
Router# configure terminal
! Locks configuration mode exclusively.

Router(config)# show configuration lock

Parser Configure Lock

Owner PID         :  10
User              :  User1
TTY               :  3
Type              :  EXCLUSIVE
State             :  LOCKED
Class             :  Exposed
Count             :  0
Pending Requests  :  0
User debug info   :  0
```

Configuring an Exclusive Lock in Manual Mode: Example

The following example shows how to enable the exclusive locking feature in manual mode by using the **configuration mode exclusive manual** command. Once you have configured manual exclusive mode, you can lock the configuration mode by using the **configure terminal lock** command. In this mode, the **configure terminal** command will not automatically lock the parser configuration mode.

```
Router#
Router# configure terminal
Router(config)# configuration mode exclusive manual
Router(config)# exit

Router# configure terminal lock
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#

*Mar 25 17:02:45.928:  Configuration mode locked exclusively. The lock will be cleared
once you exit out of configuration mode using end/exit
```

Additional References

The following sections provide references related to the Exclusive Configuration Change Access and Access Session Locking feature.

Related Documents

Related Topic	Document Title
Commands for managing configuration files	*Cisco IOS Configuration Management Command Reference*
Information about managing configuration files	Managing Configuration Files

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

- **configuration mode exclusive**
- **configure terminal**
- **debug configuration lock**
- **show configuration lock**

Feature Information for Exclusive Configuration Change Access and Access Session Locking

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Exclusive Configuration Change Access and Access Session Locking

Feature Name	Releases	Feature Information
Exclusive Configuration Change Access and Access Session Locking	12.3(14)T 12.0(31)S 12.2(33)SRA 12.4(11)T 12.2(33)SXH 12.2(33)SB Cisco IOS XE Release 2.1	Exclusive Configuration Change Access (also called the “Configuration Lock” feature) allows you to have exclusive change access to the Cisco IOS running configuration, preventing multiple users from making concurrent configuration changes. The Access Session Locking addition to this feature extends the Exclusive Configuration Change Access feature such that **show** and **debug** commands entered by the user holding the configuration lock always have execution priority; **show** and **debug** commands entered by other users are only allowed to run after the processes initiated by the configuration lock owner have finished. The Exclusive Configuration Change Access feature (“exposed lock”) is complementary with the locking mechanism in the Configuration Replace and Configuration Rollback feature (“rollback lock”). The Configuration Lock feature was integrated into Release 12.0S, and the Access Session Locking feature extension was implemented. The **configuration mode exclusive command** was extended to include the following keyword options: **expire**, **lock-show**, **interleave**, **terminate**, **config_wait**, and **retry_wait**. The output of the **show configuration lock** command was improved. The extended feature was integrated into Releases 12.2(33)SRA, 12.4(11)T, 12.2(33)SXH, and 12.2(33)SB. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following sections provide information about this feature: • Information About Exclusive Configuration Change Access and Access Session Locking • How to Use Exclusive Configuration Change Access and Access Session Locking

Configuration Replace and Configuration Rollback

First Published: March 3, 2004
Last Updated: July 11, 2008

The Configuration Replace and Configuration Rollback feature provides the capability to replace the current running configuration with any saved Cisco IOS configuration file. This functionality can be used to revert to a previous configuration state, effectively rolling back any configuration changes that were made since that configuration file was saved.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuration Replace and Configuration Rollback" section on page 17.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuration Replace and Configuration Rollback

- The format of the configuration files used as input by the Configuration Replace and Configuration Rollback feature must comply with standard Cisco IOS software configuration file indentation rules as follows:
  - Start all commands on a new line with no indentation, unless the command is within a configuration submode.
  - Indent commands within a first-level configuration submode one space.
  - Indent commands within a second-level configuration submode two spaces.
  - Indent commands within subsequent submodes accordingly.

  These indentation rules describe how Cisco IOS software creates configuration files for such Cisco IOS commands as **show running-config** or **copy running-config** *destination-url*. Any configuration file generated on a Cisco IOS device complies with these rules.
- Free memory larger than the combined size of the two configuration files (the current running configuration and the saved replacement configuration) is required.

Restrictions for Configuration Replace and Configuration Rollback

- If the router does not have free memory larger than the combined size of the two configuration files (the current running configuration and the saved replacement configuration), the configuration replace operation is not performed.
- Certain Cisco IOS configuration commands such as those pertaining to physical components of a networking device (for example, physical interfaces) cannot be added or removed from the running configuration. To illustrate, a configuration replace operation cannot remove the **interface ethernet 0** command line from the current running configuration if that interface is physically present on the device. Similarly, the **interface ethernet 1** command line cannot be added to the running configuration if no such interface is physically present on the device. A configuration replace operation that attempts to perform these types of changes results in error messages indicating that these specific command lines failed.
- In very rare cases, certain Cisco IOS configuration commands cannot be removed from the Cisco IOS running configuration without reloading the router. A configuration replace operation that attempts to remove this type of command results in error messages indicating that these specific command lines failed.

Information About Configuration Replace and Configuration Rollback

To use the Configuration Replace and Configuration Rollback feature, you should understand the following concepts:

- Configuration Archive, page 3
- Configuration Replace, page 3
- Configuration Rollback, page 4
- Benefits of Configuration Replace and Configuration Rollback, page 5

Configuration Archive

The Cisco IOS configuration archive is intended to provide a mechanism to store, organize, and manage an archive of Cisco IOS configuration files to enhance the configuration rollback capability provided by the **configure replace** command. Before this feature was introduced, you could save copies of the running configuration using the **copy running-config** *destination-url* command, storing the replacement file either locally or remotely. However, this method lacked any automated file management. On the other hand, the Configuration Replace and Configuration Rollback feature provides the capability to automatically save copies of the running configuration to the Cisco IOS configuration archive. These archived files serve as checkpoint configuration references and can be used by the **configure replace** command to revert to previous configuration states.

The **archive config** command allows you to save Cisco IOS configurations in the configuration archive using a standard location and filename prefix that is automatically appended with an incremental version number (and optional timestamp) as each consecutive file is saved. This functionality provides a means for consistent identification of saved Cisco IOS configuration files. You can specify how many versions of the running configuration are kept in the archive. After the maximum number of files are saved in the archive, the oldest file is automatically deleted when the next, most recent file is saved. The **show archive** command displays information for all configuration files saved in the Cisco IOS configuration archive.

The Cisco IOS configuration archive, in which the configuration files are stored and available for use with the **configure replace** command, can be located on the following file systems:

- If your platform has disk0—disk0:, disk1:, ftp:, pram:, rcp:, slavedisk0:, slavedisk1:, or tftp:
- If your platform does not have disk0—ftp:, http:, pram:, rcp:, or tftp:

Configuration Replace

The **configure replace** command provides the capability to replace the current running configuration with any saved Cisco IOS configuration file. This functionality can be used to revert to a previous configuration state, effectively rolling back any configuration changes that were made since the previous configuration state was saved.

When using the **configure replace** command, you must specify a saved Cisco IOS configuration as the replacement configuration file for the current running configuration. The replacement file must be a complete configuration generated by a Cisco IOS device (for example, a configuration generated by the **copy running-config** *destination-url* command), or, if generated externally, the replacement file must comply with the format of files generated by Cisco IOS devices. When the **configure replace** command is entered, the current running configuration is compared with the specified replacement configuration and a set of diffs is generated. The algorithm used to compare the two files is the same as that employed by the **show archive config differences** command. The resulting diffs are then applied by the Cisco IOS parser to achieve the replacement configuration state. Only the diffs are applied, avoiding potential service disruption from reapplying configuration commands that already exist in the current running configuration. This algorithm effectively handles configuration changes to order-dependent commands (such as access lists) through a multiple pass process. Under normal circumstances, no more than three passes are needed to complete a configuration replace operation, and a limit of five passes is performed to preclude any looping behavior.

The Cisco IOS **copy** *source-url* **running-config** command is often used to copy a stored Cisco IOS configuration file to the running configuration. When using the **copy** *source-url* **running-config** command as an alternative to the **configure replace** *target-url* command, the following major differences should be noted:

- The **copy** *source-url* **running-config** command is a merge operation and preserves all the commands from both the source file and the current running configuration. This command does not remove commands from the current running configuration that are not present in the source file. In contrast, the **configure replace** *target-url* command removes commands from the current running configuration that are not present in the replacement file and adds commands to the current running configuration that need to be added.
- The **copy** *source-url* **running-config** command applies every command in the source file, whether or not the command is already present in the current running configuration. This algorithm is inefficient and, in some cases, can result in service outages. In contrast, the **configure replace** *target-url* command only applies the commands that need to be applied—no existing commands in the current running configuration are reapplied.
- A partial configuration file may be used as the source file for the **copy** *source-url* **running-config** command, whereas a complete Cisco IOS configuration file must be used as the replacement file for the **configure replace** *target-url* command.

Note In Cisco IOS Release 12.2(25)S and 12.3(14)T, a locking feature for the configuration replace operation was introduced. When the **configure replace** command is used, the running configuration file is locked by default for the duration of the configuration replace operation. This locking mechanism prevents other users from changing the running configuration while the replacement operation is taking place, which might otherwise cause the replacement operation to terminate unsuccessfully. You can disable the locking of the running configuration by using the **nolock** keyword when issuing the **configure replace** command.

The running configuration lock is automatically cleared at the end of the configuration replace operation. You can display any locks that may be currently applied to the running configuration using the **show configuration lock** command.

Configuration Rollback

The concept of rollback comes from the transactional processing model common to database operations. In a database transaction, you might make a set of changes to a given database table. You then must choose whether to commit the changes (apply the changes permanently) or to roll back the changes (discard the changes and revert to the previous state of the table). In this context, rollback means that a journal file containing a log of the changes is discarded, and no changes are applied. The result of the rollback operation is to revert to the previous state, before any changes were applied.

The **configure replace** command allows you to revert to a previous configuration state, effectively rolling back changes that were made since the previous configuration state was saved. Instead of basing the rollback operation on a specific set of changes that were applied, the Cisco IOS configuration rollback capability uses the concept of reverting to a specific configuration state based on a saved Cisco IOS configuration file. This concept is similar to the database idea of saving a checkpoint (a saved version of the database) to preserve a specific state.

If the configuration rollback capability is desired, you must save the Cisco IOS running configuration before making any configuration changes. Then, after entering configuration changes, you can use that saved configuration file to roll back the changes (using the **configure replace** *target-url* command). Furthermore, since you can specify any saved Cisco IOS configuration file as the replacement configuration, you are not limited to a fixed number of rollbacks, as is the case in some rollback models based on a journal file.

Configuration Rollback Confirmed Change

The Configuration Rollback Confirmed Change feature enables an added criteria of a confirmation to configuration changes. This functionality enables a rollback to occur if a confirmation of the requested changes is not received in a configured time frame. Command failures can also be configured to trigger a configuration rollback.

The following steps outline how this process is achieved:

1. When entering configuration mode, this new option allows you to request confirmation (a confirmation time limit must be supplied) of the configuration changes.
2. After exiting configuration mode, you must enter the confirmation command. If no confirmation is entered within the requested time limit, the configuration will revert to its previous state.

Benefits of Configuration Replace and Configuration Rollback

- Allows you to revert to a previous configuration state, effectively rolling back configuration changes.
- Allows you to replace the current running configuration file with the startup configuration file without having to reload the router or manually undo CLI changes to the running configuration file, therefore reducing system downtime.
- Allows you to revert to any saved Cisco IOS configuration state.
- Simplifies configuration changes by allowing you to apply a complete configuration file to the router, where only the commands that need to be added or removed are affected.
- When using the **configure replace** command as an alternative to the **copy** *source-url* **running-config** command, increases efficiency and prevents risk of service outages by not reapplying existing commands in the current running configuration.

How to Use Configuration Replace and Configuration Rollback

This section contains the following procedures:

- Creating a Configuration Archive, page 6 (optional)
- Performing a Configuration Replace or Configuration Rollback Operation, page 7 (required)

- Monitoring and Troubleshooting the Configuration Replace and Configuration Rollback Feature, page 9 (optional)

Creating a Configuration Archive

No prerequisite configuration is needed to use the **configure replace** command. Using the **configure replace** command in conjunction with the Cisco IOS configuration archive and the **archive config** command is optional but offers significant benefit for configuration rollback scenarios. Before using the **archive config** command, the configuration archive must be configured. Perform this task to configure the characteristics of the configuration archive.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **archive**
4. **path** *url*
5. **maximum** *number*
6. **time-period** *minutes*
7. **end**
8. **archive config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`archive` **Example:** `Router(config)# archive`	Enters archive configuration mode.
Step 4	`path` *url* **Example:** `Router(config-archive)# path disk0:myconfig`	Specifies the location and filename prefix for the files in the Cisco IOS configuration archive. • The *url* argument is a URL (accessible by the Cisco IOS file system) used for saving archive files of the running configuration file in the Cisco IOS configuration archive. You can set up an archive on any file system that your platform supports (see the "Configuration Archive" section on page 3).

	Command or Action	Purpose
Step 5	`maximum number` **Example:** `Router(config-archive)# maximum 14`	(Optional) Sets the maximum number of archive files of the running configuration to be saved in the Cisco IOS configuration archive. • The *number* argument is the maximum number of archive files of the running configuration to be saved in the Cisco IOS configuration archive. Valid values are from 1 to 14. The default is 10. **Note** Before using this command, you must configure the **path** command to specify the location and filename prefix for the files in the Cisco IOS configuration archive.
Step 6	`time-period minutes` **Example:** `Router(config-archive)# time-period 10`	(Optional) Sets the time increment for automatically saving an archive file of the current running configuration in the Cisco IOS configuration archive. • The minutes argument specifies how often, in minutes, to automatically save an archive file of the current running configuration in the Cisco IOS configuration archive. **Note** Before using this command, you must configure the **path** command to specify the location and filename prefix for the files in the Cisco IOS configuration archive.
Step 7	`end` **Example:** `Router(config-archive)# end`	Exits to privileged EXEC mode.
Step 8	`archive config` **Example:** `Router# archive config`	Saves the current running configuration file to the configuration archive. **Note** The **path** command must be configured before using this command.

Performing a Configuration Replace or Configuration Rollback Operation

Perform this task to replace the current running configuration file with a saved Cisco IOS configuration file.

Note You must create a configuration archive before performing this procedure. See Creating a Configuration Archive, page 6 for detailed steps. The following procedure details how to return to that archived configuration in the event of a problem with the current running configuration.

SUMMARY STEPS

1. **enable**
2. **configure replace** *target-url* [**nolock**] [**list**] [**force**] [**ignorecase**] [**revert trigger** [**error**] [**timer** *minutes*] | **time** *minutes*]

3. **configure revert** {**now** | **timer** {*minutes* | **idle** *minutes*}}
4. **configure confirm**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure replace`** *`target-url`* [**`nolock`**] [**`list`**] [**`force`**] [**`ignorecase`**] [**`revert trigger`** [**`error`**] [**`timer`** *`minutes`*] | **`time`** *`minutes`*] **Example:** `Router# configure replace disk0:myconfig-1 list time 30`	Replaces the current running configuration file with a saved Cisco IOS configuration file. • The *target-url* argument is a URL (accessible by the Cisco IOS file system) of the saved Cisco IOS configuration file that is to replace the current running configuration, such as the configuration file created using the **archive config** command. • The **list** keyword displays a list of the command lines applied by the Cisco IOS software parser during each pass of the configuration replace operation. The total number of passes performed is also displayed. • The **force** keyword replaces the current running configuration file with the specified saved Cisco IOS configuration file without prompting you for confirmation. • The **time** *minutes* keyword and argument specify the time (in minutes) within which you must enter the **configure confirm** command to confirm replacement of the current running configuration file. If the **configure confirm** command is not entered within the specified time limit, the configuration replace operation is automatically reversed (in other words, the current running configuration file is restored to the configuration state that existed prior to entering the **configure replace** command). • The **nolock** keyword disables the locking of the running configuration file that prevents other users from changing the running configuration during a configuration replace operation. • The **revert trigger** keywords set the following triggers for reverting to the original configuration: – **error**—Reverts to the original configuration upon error. – **timer** *minutes*—Reverts to the original configuration if specified time elapses. • The **ignorecase** keyword allows the configuration to ignore the case of the confirmation command.

	Command or Action	Purpose
Step 3	**configure revert** {**now** | **timer** {*minutes* | **idle** *minutes*}} **Example:** `Router# configure revert now`	(Optional) To cancel the timed rollback and trigger the rollback immediately, or to reset parameters for the timed rollback, use the **configure revert** command in privileged EXEC mode. • **now**—Triggers the rollback immediately. • **timer**—Resets the configuration revert timer. – Use the *minutes* argument with the **timer** keyword to specify a new revert time in minutes. – Use the **idle** keyword along with a time in minutes to set the maximum allowable time period of no activity before reverting to the saved configuration.
Step 4	**configure confirm** **Example:** `Router# configure confirm`	(Optional) Confirms replacement of the current running configuration file with a saved Cisco IOS configuration file. **Note** Use this command only if the **time** *seconds* keyword and argument of the **configure replace** command are specified.
Step 5	**exit** **Example:** `Router# exit`	Exits to user EXEC mode.

Monitoring and Troubleshooting the Configuration Replace and Configuration Rollback Feature

Perform this task to monitor and troubleshoot the Configuration Replace and Configuration Rollback feature.

SUMMARY STEPS

1. **enable**
2. **show archive**
3. **debug archive versioning**
4. **debug archive config timestamp**
5. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

```
Router> enable
Router#
```

Step 2 **show archive**

Use this command to display information about the files saved in the Cisco IOS configuration archive. For example:

```
Router# show archive

There are currently 1 archive configurations saved.
The next archive file will be named disk0:myconfig-2
 Archive #  Name
   0
   1       disk0:myconfig-1 <- Most Recent
   2
   3
   4
   5
   6
   7
   8
   9
   10
   11
   12
   13
   14
```

The following is sample output from the **show archive** command after several archive files of the running configuration have been saved. In this example, the maximum number of archive files to be saved is set to three.

```
Router# show archive

There are currently 3 archive configurations saved.
The next archive file will be named disk0:myconfig-8
 Archive #  Name
   0
   1       :Deleted
   2       :Deleted
   3       :Deleted
   4       :Deleted
   5       disk0:myconfig-5
   6       disk0:myconfig-6
   7       disk0:myconfig-7 <- Most Recent
   8
   9
   10
   11
   12
   13
   14
```

Step 3 **debug archive versioning**

Use this command to enable debugging of the Cisco IOS configuration archive activities to help monitor and troubleshoot configuration replace and rollback. For example:

```
Router# debug archive versioning

Jan  9 06:46:28.419:backup_running_config
Jan  9 06:46:28.419:Current = 7
Jan  9 06:46:28.443:Writing backup file disk0:myconfig-7
Jan  9 06:46:29.547: backup worked
```

Step 4 **debug archive config timestamp**

Use this command to enable debugging of the processing time for each integral step of a configuration replace operation and the size of the configuration files being handled. For example:

```
Router# debug archive config timestamp
Router# configure replace disk0:myconfig force

Timing Debug Statistics for IOS Config Replace operation:
       Time to read file slot0:sample_2.cfg = 0 msec (0 sec)
       Number of lines read:55
       Size of file        :1054

Starting Pass 1
       Time to read file system:running-config = 0 msec (0 sec)
       Number of lines read:93
       Size of file        :2539
       Time taken for positive rollback pass = 320 msec (0 sec)
       Time taken for negative rollback pass = 0 msec (0 sec)
       Time taken for negative incremental diffs pass = 59 msec (0 sec)
       Time taken by PI to apply changes = 0 msec (0 sec)
       Time taken for Pass 1 = 380 msec (0 sec)
Starting Pass 2
       Time to read file system:running-config = 0 msec (0 sec)
       Number of lines read:55
       Size of file        :1054
       Time taken for positive rollback pass = 0 msec (0 sec)
       Time taken for negative rollback pass = 0 msec (0 sec)
       Time taken for Pass 2 = 0 msec (0 sec)

Total number of passes:1
Rollback Done
```

Step 5 **exit**

Use this command to exit to user EXEC mode. For example:

```
Router# exit
Router>
```

Configuration Examples for Configuration Replace and Configuration Rollback

This section provides the following configuration examples:

Creating a Configuration Archive: Example

The following example shows how to perform the initial configuration of the Cisco IOS configuration archive. In this example, disk0:myconfig is specified as the location and filename prefix for the files in the configuration archive and a value of 10 is set as the maximum number of archive files to be saved.

```
configure terminal
!
archive
 path disk0:myconfig
 maximum 10
 end
```

Replacing the Current Running Configuration with a Saved Cisco IOS Configuration File: Example

The following example shows how to replace the current running configuration with a saved Cisco IOS configuration file named disk0:myconfig. The **configure replace** command interactively prompts you to confirm the operation.

```
Router# configure replace disk0:myconfig

This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y

Total number of passes: 1
Rollback Done
```

In the following example, the **list** keyword is specified in order to display the command lines that were applied during the configuration replace operation:

```
Router# configure replace disk0:myconfig list

This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y

!Pass 1

!List of Commands:
no snmp-server community public ro
snmp-server community mystring ro
end

Total number of passes: 1
Rollback Done
```

Reverting to the Startup Configuration File: Example

The following example shows how to revert to the Cisco IOS startup configuration file using the **configure replace** command. This example also shows the use of the optional **force** keyword to override the interactive user prompt.

```
Router# configure replace nvram:startup-config force

Total number of passes: 1
Rollback Done
```

Performing a Configuration Replace Operation with the configure confirm Command: Example

The following example shows the use of the **configure replace** command with the **time** *seconds* keyword and argument. You must enter the **configure confirm** command within the specified time limit to confirm replacement of the current running configuration file. If the **configure confirm** command is not entered within the specified time limit, the configuration replace operation is automatically reversed (in other words, the current running configuration file is restored back to the configuration state that existed prior to entering the **configure replace** command).

```
Router# configure replace nvram:startup-config time 120

This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y

Total number of passes: 1
Rollback Done

Router# configure confirm
```

Performing a Configuration Rollback Operation: Example

The following example shows how to make changes to the current running configuration and then roll back the changes. As part of the configuration rollback operation, you must save the current running configuration before making changes to the file. In this example, the **archive config** command is used to save the current running configuration. The generated output of the **configure replace** command indicates that only one pass was performed to complete the rollback operation.

Note Before using the **archive config** command, you must configure the **path** command to specify the location and filename prefix for the files in the Cisco IOS configuration archive.

You first save the current running configuration in the configuration archive as follows:

```
archive config
```

You then enter configuration changes as shown in the following example:

```
configure terminal
!
user netops2 password rain
user netops3 password snow
exit
```

After having made changes to the running configuration file, assume you now want to roll back these changes and revert to the configuration that existed before the changes were made. The **show archive** command is used to verify the version of the configuration to be used as a replacement file. The **configure replace** command is then used to revert to the replacement configuration file as shown in the following example:

```
Router# show archive

There are currently 1 archive configurations saved.
The next archive file will be named disk0:myconfig-2
 Archive #  Name
   0
   1       disk0:myconfig-1 <- Most Recent
   2
   3
   4
   5
   6
   7
   8
   9
   10

Router# configure replace disk0:myconfig-1

Total number of passes: 1
Rollback Done
```

Additional References

The following sections provide references related to the Configuration Replace and Configuration Rollback feature.

Related Documents

Related Topic	Document Title
Configuration Locking	Exclusive Configuration Change Access and Access Session Locking
Commands for managing configuration files	*Cisco IOS Configuration Fundamentals Command Reference,*
Information about managing configuration files	Managing Configuration Files
Using the Contextual Configuration Diff Utility feature	Contextual Configuration Diff Utility

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **archive config**
- **configure confirm**
- **configure replace**
- **configure revert**
- **configure terminal**
- **debug archive config timestamp**
- **debug archive versioning**
- **maximum**
- **path (archive configuration)**
- **show archive**
- **show configuration lock**
- **time-period**

Feature Information for Configuration Replace and Configuration Rollback

Table 1 lists the release history for this feature. Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release. Unless noted otherwise, subsequent releases of that Cisco IOS software release also support that feature.

Table 1 Feature Information for Configuration Replace and Configuration Rollback

Feature Name	Releases	Feature Information
Configuration Replace and Configuration Rollback	12.3(7)T 12.2(25)S 12.3(14)T 12.2(27)SBC 12.2(31)SB2 12.2(33)SRA 12.2(33)SXH 12.2(33)SB Cisco IOS XE Release 2.1	The Configuration Replace and Configuration Rollback feature provides the capability to replace the current running configuration with any saved Cisco IOS configuration file. This functionality can be used to revert to a previous configuration state, rolling back any configuration changes that were made since that configuration file was saved. In 12.3(7)T, this feature was introduced. In 12.2(25)S, support was added for a Cisco IOS 12.2S release. A locking mechanism for configuration replace (the Exclusive Configuration Change Access feature) was introduced. In 12.3(14)T, support for a locking mechanism for configuration replace (the Exclusive Configuration Change Access feature) was added for a Cisco IOS 12.3T release. In 12.2(27)SBC, support was added for a Cisco IOS 12.2SB release. In 12.2(33)SRA, support was added for a Cisco IOS 12.2SR release. In 12.2(31)SB2, this feature was implemented on the Cisco 10000 series. In 12.2(33)SXH, the "Configuration Rollback" feature was implemented in Release 12.2SX. In 12.2(33)SB, this feature was implemented on the Cisco 10000 series. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following sections provide feature information: • Configuration Archive, page 3 • Configuration Replace, page 3 • Configuration Rollback, page 4 • Benefits of Configuration Replace and Configuration Rollback, page 5 • Creating a Configuration Archive, page 6 • Performing a Configuration Replace or Configuration Rollback Operation, page 7 • Monitoring and Troubleshooting the Configuration Replace and Configuration Rollback Feature, page 9 The following commands were modified by this feature: **archive config**, **configure confirm**, **configure replace**, **debug archive config timestamp**, **debug archive versioning**, **maximum**, **path (archive configuration)**, **show archive**, **show configuration lock**, **time-period**.

Table 1 Feature Information for Configuration Replace and Configuration Rollback (continued)

Feature Name	Releases	Feature Information
Configuration Versioning	12.3(7)T 12.2(25)S 12.2(33)SRA Cisco IOS XE Release 2.1	The Configuration Versioning feature allows you to maintain and manage backup copies of the Cisco IOS running configuration on or off the device. The Configuration Replace feature uses the Configuration Versioning feature to provide a rollback to a saved copy of the running configuration. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers.
Exclusive Configuration Change Access	12.3(14)T 12.0(31)S 12.2(33)SRA 12.4(11)T 12.2(33)SXH 12.2(33)SB Cisco IOS XE Release 2.1	The Exclusive Configuration Change Access feature (also called the "Configuration Lock" feature) allows you to have exclusive change access to the Cisco IOS running configuration, preventing multiple users from making concurrent configuration changes. The following command was modified by this feature and applies to the Configuration Replace and Configuration Rollback feature: **show configuration lock**. Refer to the separate module, Exclusive Configuration Change Access and Access Session Locking, for details
Configuration Rollback Confirmed Change	12.2(33)SRC 12.2(33)SB Cisco IOS XE Release 2.1 12.4(20)T	The Configuration Rollback Confirmed Change feature allows configuration changes to be performed with an optional requirement that they be confirmed. If this confirmation is not received, the configuration is returned to the state prior to the changes being applied. This mechanism provides a safeguard against inadvertent loss of connectivity between a network device and the user or management application due to configuration changes. In 12.2(33)SB, this feature was implemented on the Cisco 10000 series. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following sections provide information about this feature: • Configuration Rollback Confirmed Change, page 5 • Performing a Configuration Replace or Configuration Rollback Operation, page 7 The following commands were modified by this feature: **configure confirm**, **configure replace**, **configure revert**, **configure terminal**

Configuration Change Notification and Logging

First Published: November 3, 2003
Last Updated: May 2, 2008

Prior to the introduction of this feature, the only way to determine if the Cisco IOS software configuration had changed was to save a copy of the running and startup configurations to a local computer and do a line-by-line comparison. This comparison method can identify changes that occurred, but does not specify the sequence in which the changes occurred, or the person responsible for the changes.

The Configuration Change Notification and Logging (Config Log Archive) feature allows the tracking of configuration changes entered on a per-session and per-user basis by implementing an archive function. This archive saves 'configuration logs' that track each configuration command that is applied, who applied the command, the parser return code (PRC) for the command, and the time the command was applied. This feature also adds a notification mechanism that sends asynchronous notifications to registered applications whenever the configuration log changes.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuration Change Notification and Logging" section on page 12.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for Configuration Change Notification and Logging

- Only complete commands input in a configuration mode are logged.
- Commands that are part of a configuration file applied with the **copy** command are not logged.

Information About Configuration Change Notification and Logging

To configure the Configuration Change Notification and Logging feature, you must understand the following concepts:

- Configuration Log, page 2
- Configuration Change Notifications and Config Change Logging, page 3

Configuration Log

The Configuration Change Notification and Logging feature tracks changes made to the Cisco IOS software running configuration by maintaining a configuration log. This configuration log tracks changes initiated only through the command-line interface (CLI) or HTTP. Only complete commands that result in the invocation of action routines are logged. The following types of entries are not logged:

- Commands that result in a syntax error message
- Partial commands that invoke the router help system

For each configuration command that is executed, the following information is logged:

- The command that was executed
- The configuration mode in which the command was executed
- The name of the user that executed the command
- The time at which the command was executed
- A configuration change sequence number
- Parser return codes for the command

You can display information from the configuration log through the use of the **show archive log config** command, with the exception of the parser return codes, which are for use by internal Cisco IOS applications only.

Configuration Change Notifications and Config Change Logging

You can configure the Configuration Change and Notification Logging feature to send notification of configuration changes to the Cisco IOS software system logging (syslog) process. Syslog notifications allow monitoring of the configuration log information without performing polling and information gathering tasks.

The Configuration Change Notification and Logging feature allows the tracking of configuration changes entered by users on a per-session and per-user basis. This tool allows administrators to track any configuration change made to the Cisco IOS software running configuration, and identify the user that made that change.

Config Logger Enhancements for EAL4+ Certification

Further enhancements to the Configuration Change Logging process were implemented in Cisco IOS Release 12.3(14)T. These enhancements support an effort to ensure the logging process meets the requirements set forth in the Conformance to Common Criteria, Evaluation Assurance Level 4+ (EAL4+) Firewall Protection Profiles. These enhancements include changes to meet the following requirements:

- If you change any logging parameters, those changes are logged. This is effected by the sending of a syslog message for each change to the running-config from a copy operation (for example, on **copy** *source* **running-config**).
- Modifications to the Group of Administrative Users are logged; failure attempts for access to privileged EXEC mode ("enable" mode) are logged.

Note EAL Certification is not claimed by Cisco for Cisco IOS Release 12.3(14)T. These enhancements provide the groundwork for future Certification.

The above logging actions are disabled by default. To enable these logging characteristics, perform the task described in the "Configuring the Configuration Change Notification and Logging Feature" section on page 4.

How to Configure the Configuration Change Notification and Logging Feature

This section contains the following procedures:

Configuring the Configuration Change Notification and Logging Feature

Perform this task to enable the Configuration Change Notification and Logging feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **archive**
4. **log config**
5. **logging enable**
6. **logging size** *entries*
7. **hidekeys**
8. **notify syslog**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`archive`** **Example:** `Router(config)# archive`	Enters archive configuration mode.
Step 4	**`log config`** **Example:** `Router(config-archive)# log config`	Enters configuration change logger configuration mode.
Step 5	**`logging enable`** **Example:** `Router(config-archive-log-config)# logging enable`	Enables the logging of configuration changes. • Logging of configuration changes is disabled by default.

	Command or Action	Purpose
Step 6	**logging size** *entries* **Example:** `Router(config-archive-log-config)# logging size 200`	(Optional) Specifies the maximum number of entries retained in the configuration log. • Valid values for the *entries* argument range from 1 to 1000. The default value is 100 entries. • When the configuration log is full, the oldest entry is deleted every time a new entry is added. **Note** If a new log size is specified that is smaller than the current log size, the oldest log entries is immediately purged until the new log size is satisfied, regardless of the age of the log entries.
Step 7	**hidekeys** **Example:** `Router(config-archive-log-config)# hidekeys`	(Optional) Suppresses the display of password information in configuration log files. **Note** Enabling the **hidekeys** command increases security by preventing password information from being displayed in configuration log files.
Step 8	**notify syslog** **Example:** `Router(config-archive-log-config)# notify syslog`	(Optional) Enables the sending of notifications of configuration changes to a remote syslog.
Step 9	**end** **Example:** `Router(config-archive-log-config)# end`	Exits to privileged EXEC mode.

Displaying Configuration Log Entries and Statistics

Perform this task to display entries from the configuration log or statistics about the memory usage of the configuration log.

To display configuration log entries and to monitor the memory usage of the configuration log, the Configuration Change Notification and Logging feature provides the **show archive log config** command.

SUMMARY STEPS

1. **enable**
2. **show archive log config** *number* [*end-number*]
3. **show archive log config all provisioning**
4. **show archive log config statistics**
5. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example”

```
Router> enable
```

Step 2 **show archive log config** *number* [*end-number*]

Use this command to display configuration log entries by record numbers. If you specify a record number for the optional *end-number* argument, all log entries with record numbers between the values entered for the *number* and *end-number* arguments are displayed. For example:

```
Router# show archive log config 1 2

idx   sess   user@line        Logged command
 1     1     user1@console    logging enable
 2     1     user1@console    logging size 200
```

This example displays configuration log entry numbers 1 and 2. Valid values for the *number* and *end-number* argument range from 1 to 2147483647.

Step 3 **show archive log config provisioning**

Use this command to display all configuration log files as they would appear in a configuration file rather than in tabular format. For example:

```
Router# show archive log config all provisioning

archive
 log config
  logging enable
  logging size 200
```

This display also shows the commands used to change configuration modes, which are required to correctly apply the logged commands.

Step 4 **show archive log config statistics**

Use this command to display memory usage information for the configuration. For example:

```
Router# show archive log config statistics

Config Log Session Info:
   Number of sessions being tracked: 1
   Memory being held: 3910 bytes
   Total memory allocated for session tracking: 3910 bytes
   Total memory freed from session tracking: 0 bytes

Config Log log-queue Info:
   Number of entries in the log-queue: 3
   Memory being held in the log-queue: 671 bytes
   Total memory allocated for log entries: 671 bytes
   Total memory freed from log entries:: 0 bytes
```

Step 5 **exit**

Use this command to exit to user EXEC mode. For example:

```
Router# exit
Router>
```

Clearing Configuration Log Entries

Entries from the configuration log can be cleared in one of two ways. The size of the configuration log can be reduced using the **logging size** command, or the configuration log can be disabled and then reenabled with the **logging enable** command.

This section contains the following procedures:

- Clearing the Configuration Log by Reducing the Log Size, page 7
- Clearing the Configuration Log by Disabling the Configuration Log, page 8

Clearing the Configuration Log by Reducing the Log Size

Perform this task to clear entries from the configuration log using the **logging size** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **archive**
4. **log config**
5. **logging size** *entries*
6. **logging size** *entries*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`archive`** **Example:** `Router(config)# archive`	Enters archive configuration mode.
Step 4	**`log config`** **Example:** `Router(config-archive)# log config`	Enters configuration change logger configuration mode.

	Command or Action	Purpose
Step 5	**`logging size`** *`entries`* **Example:** `Router(config-archive-log-config)# logging size 1`	Specifies the maximum number of entries retained in the configuration log. **Note** Setting the size of the configuration log to 1 results in all but the most recent entry being purged.
Step 6	**`logging size`** *`entries`* **Example:** `Router(config-archive-log-config)# logging size 200`	Specifies the maximum number of entries retained in the configuration log. **Note** The size of the configuration log should be reset to the desired value after clearing the configuration log.
Step 7	**`end`** **Example:** `Router(config-archive-log-config)# end`	Exits to privileged EXEC mode.

Examples

The following example shows how to clear the configuration log by reducing the log size to 1, then resetting the log size to the desired value:

```
Router# configure terminal

Router(config)# archive
Router(config-archive)# log config
Router(config-archive-log-config)# logging size 1
Router(config-archive-log-config)# logging size 200
Router(config-archive-log-config)# end
```

Clearing the Configuration Log by Disabling the Configuration Log

Perform this task to clear entries from the configuration log using the **logging enable** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **archive**
4. **log config**
5. **no logging enable**
6. **logging enable**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**archive** **Example:** `Router(config)# archive`	Enters archive configuration mode.
Step 4	**log config** **Example:** `Router(config-archive)# log config`	Enters configuration change logger configuration mode.
Step 5	**no logging enable** **Example:** `Router(config-archive-log-config)# no logging enable`	Disables the logging of configuration changes. **Note** Disabling the configuration log results in all records being purged.
Step 6	**logging enable** **Example:** `Router(config-archive-log-config)# logging enable`	Enables the logging of configuration changes.
Step 7	**end** **Example:** `Router(config-archive-log-config)# end`	Exits to privileged EXEC mode.

Examples

The following example clears the configuration log by disabling and then reenabling the configuration log:

```
Router(config)# archive
Router(config-archive)# log config
Router(config-archive-log-config)# no logging enable
Router(config-archive-log-config)# logging enable
Router(config-archive-log-config)# end
```

Configuration Examples for the Configuration Change Notification and Logging Feature

This section provides the following configuration example:

- Configuring the Configuration Change Notification and Logging Feature: Example

Configuring the Configuration Change Notification and Logging Feature: Example

The following example shows how to enable configuration logging with a maximum of 200 entries in the configuration log. In the example, security is increased by suppressing the display of password information in configuration log records, and syslog notifications are turned on.

```
configure terminal

archive
 log config
 logging enable
 logging size 200
 hidekeys
 notify syslog
```

Additional References

The following sections provide references related to the Configuration Change Notification and Logging. feature:

Related Documents

Related Topic	Document Title
Information about managing configuration files	Managing Configuration Files
Commands for managing configuration files	*Cisco IOS Configuration Fundamentals Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **archive**
- **hidekeys**
- **log config**
- **logging enable (config-archive-log)**
- **logging size (config-archive-log)**

- **notify syslog**
- **show archive log config**

Feature Information for Configuration Change Notification and Logging

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Configuration Change Notification and Logging

Feature Name	Releases	Feature Information
Configuration Change Notification and Logging	12.3(4)T 12.2(25)S 12.2(27)SBC 12.2(33)SRA 12.2(33)SXH 12.2(33)SB Cisco IOS XE Release 2.1	The Configuration Change Notification and Logging (Configuration Logging) feature allows the tracking of configuration changes entered on a per-session and per-user basis by implementing a configuration log. The configuration log tracks each configuration command that is applied, who applied the command, the parser return code for the command, and the time the command was applied. This feature also adds a notification mechanism that sends asynchronous notifications to registered applications whenever the configuration log changes. In 12.2(33)SB, this feature was implemented on the Cisco 10000 series. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following sections provide information about this feature: • Configuration Change Notifications and Config Change Logging, page 3 • Configuring the Configuration Change Notification and Logging Feature, page 4 • Displaying Configuration Log Entries and Statistics, page 5 The following commands were modified by this feature: **archive**, **hidekeys**, **log config**, **logging enable**, **logging size**, **notify syslog**, **show archive log config**.
Config Logger Enhancements for EAL4+ Certification	12.3(14)T 12.2(27)SBC	Further enhancements to the Configuration Change Logging process were implemented in Cisco IOS Release 12.3(14)T and 12.2(27)SBC. These enhancements support an effort to ensure the logging process meets the requirements set forth in the Conformance to Common Criteria, Evaluation Assurance Level 4+ (EAL4+) Firewall Protection Profiles. The following section provides information about this feature: • Config Logger Enhancements for EAL4+ Certification, page 3

Contextual Configuration Diff Utility

First Published: November 2003
Last Updated: May 2, 2008

The Contextual Configuration Diff Utility feature provides the ability to perform a line-by-line comparison of any two configuration files (accessible through the Cisco IOS Integrated File System [IFS]) and generate a list of the differences between them. The generated output includes information regarding configuration lines that have been added, modified, or deleted, and the configuration modes within which a changed configuration line exists.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Contextual Configuration Diff Utility" section on page 8.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Contextual Configuration Diff Utility

The format of the configuration files used for the Contextual Configuration Diff Utility feature must comply with standard Cisco IOS configuration file indentation rules as follows:

- Start all commands on a new line with no indentation, unless the command is within a configuration submode.
- Indent commands within a first-level configuration submode one space.
- Indent commands within a second-level configuration submode two spaces.
- Indent commands within subsequent submodes accordingly.

The router must have a contiguous block of memory larger than the combined size of the two configuration files being compared.

Restrictions for Contextual Configuration Diff Utility

If the router does not have a contiguous block of memory larger than the combined size of the two configuration files being compared, the diff operation fails.

Information About Contextual Configuration Diff Utility

Before using the Contextual Configuration Diff Utility feature, you should understand the following concepts:

- Benefits of the Contextual Configuration Diff Utility, page 2
- Contextual Configuration Diff Utility Output Format, page 2

Benefits of the Contextual Configuration Diff Utility

The Contextual Configuration Diff Utility feature provides the ability to perform a line-by-line comparison of any two configuration files (accessible through the Cisco IOS File System [IFS]) and generate a list of the differences between them. The generated output includes information regarding the following items:

- Configuration lines that have been added, modified, or deleted.
- Configuration modes within which a changed configuration line exists.
- Location changes of configuration lines that are order-sensitive. For example, the **ip access-list** and **community-lists** commands are order-sensitive commands dependent on where they are listed within a configuration file in relation to other Cisco IOS commands of similar type.

Contextual Configuration Diff Utility Output Format

Diff Operation

The Contextual Configuration Diff Utility feature uses the filenames of two configuration files as input. A diff operation is performed on the specified files and a list of differences between the two files is generated as output. Interpreting the output is dependent on the order in which the two files are configured (**show archive config differences** command). In this section, we assume that the filename of the file entered first is file1 and the filename of the file entered second is file2. Each entry in the generated output list is prefixed with a unique text symbol to indicate the type of difference found. The text symbols and their meanings are as follows:

- A minus symbol (-) indicates that the configuration line exists in file1 but not in file2.
- A plus symbol (+) indicates that the configuration line exists in file2 but not in file1.
- An exclamation point (!) with descriptive comments is used to identify order-sensitive configuration lines whose location is different in file1 than in file2.

Incremental Diff Operation

Some applications require that the generated output of a diff operation contain configuration lines that are unmodified (in other words, without the minus and plus symbols). For these applications, an incremental diff operation can be performed, which compares a specified configuration file to the running configuration file (**show archive config incremental-diffs** command).

When an incremental diff operation is performed, a list of the configuration lines that do not appear in the running configuration file (in other words, configuration lines that only appear in the specified file that is being compared to the running configuration file) is generated as output. An exclamation point (!) with descriptive comments is used to identify order-sensitive configuration lines whose location is different in the specified configuration file than in the running configuration file.

How to Use the Contextual Configuration Diff Utility

This section provides the following procedure:

- Using the Contextual Configuration Diff Utility, page 3 (required)

Using the Contextual Configuration Diff Utility

This task describes how to use the Contextual Configuration Diff Utility feature.

SUMMARY STEPS

1. **enable**
2. **show archive config differences** [*file1* [*file2*]]
   or
   show archive config incremental-diffs [*file*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show archive config differences** [*file1* [*file2*]] or **show archive config incremental-diffs** *file* **Example:** `Router# show archive config differences running-config startup-config` or **Example:** `Router# show archive config incremental-diffs nvram:startup-config`	Performs a line-by-line comparison of any two configuration files (accessible through the IFS) and generates a list of the differences between them. or Performs a line-by-line comparison of a specified configuration file to the running configuration file and generates a list of the configuration lines that do not appear in the running configuration file.
Step 3	**exit** **Example:** `Router# exit`	Exits to user EXEC mode.

Configuration Examples for the Contextual Configuration Diff Utility

This section contains the following configuration examples:

- Diff Operation: Example, page 4
- Incremental Diff Operation: Example, page 6

Diff Operation: Example

In this example, a diff operation is performed on the running and startup configuration files. Table 1 shows the configuration files used for this example.

Table 1 Configuration Files Used for the Diff Operation Example

Running Configuration File	Startup Configuration File
no ip subnet-zero ip cef interface Ethernet1/0 ip address 10.7.7.7 255.0.0.0 no ip route-cache no ip mroute-cache duplex half no ip classless snmp-server community public RO	ip subnet-zero ip cef ip name-server 10.4.4.4 voice dnis-map 1 dnis 111 interface Ethernet1/0 no ip address no ip route-cache no ip mroute-cache shutdown duplex half ip default-gateway 10.5.5.5 ip classless access-list 110 deny ip any host 10.1.1.1 access-list 110 deny ip any host 10.1.1.2 access-list 110 deny ip any host 10.1.1.3 snmp-server community private RW

The following is sample output from the **show archive config differences** command. This sample output displays the results of the diff operation performed on the configuration files in Table 1.

```
Router# show archive config differences running-config startup-config

+ip subnet-zero
+ip name-server 10.4.4.4
+voice dnis-map 1
 +dnis 111
interface Ethernet1/0
 +no ip address
 +shutdown
+ip default-gateway 10.5.5.5
+ip classless
+access-list 110 deny ip any host 10.1.1.1
+access-list 110 deny ip any host 10.1.1.2
+access-list 110 deny ip any host 10.1.1.3
+snmp-server community private RW
-no ip subnet-zero
interface Ethernet1/0
 -ip address 10.7.7.7 255.0.0.0
-no ip classless
-snmp-server community public RO
```

Incremental Diff Operation: Example

In this example, an incremental diff operation is performed on the startup and running configuration files. Table 2 shows the configuration files used for this example.

Table 2 Configuration Files Used for the Incremental Diff Operation Example

Startup Configuration File	Running Configuration File
ip subnet-zero ip cef ip name-server 10.4.4.4 voice dnis-map 1 dnis 111 interface Ethernet1/0 no ip address no ip route-cache no ip mroute-cache shutdown duplex half ip default-gateway 10.5.5.5 ip classless access-list 110 deny ip any host 10.1.1.1 access-list 110 deny ip any host 10.1.1.2 access-list 110 deny ip any host 10.1.1.3 snmp-server community private RW	no ip subnet-zero ip cef interface Ethernet1/0 ip address 10.7.7.7 255.0.0.0 no ip route-cache no ip mroute-cache duplex half no ip classless snmp-server community public RO

The following is sample output from the **show archive config incremental-diffs** command. This sample output displays the results of the incremental diff operation performed on the configuration files in Table 2.

```
Router# show archive config incremental-diffs startup-config

ip subnet-zero
ip name-server 10.4.4.4
voice dnis-map 1
 dnis 111
interface Ethernet1/0
 no ip address
 shutdown
ip default-gateway 10.5.5.5
ip classless
 access-list 110 deny   ip any host 10.1.1.1
 access-list 110 deny   ip any host 10.1.1.2
 access-list 110 deny   ip any host 10.1.1.3
snmp-server community private RW
```

Additional References

This section provides references related to the Contextual Configuration Diff Utility feature.

Related Documents

Related Topic	Document Title
Information about managing configuration files	Managing Configuration Files
Commands for managing configuration files	The *Cisco IOS Configuration Fundamentals Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **show archive config differences**
- **show archive config incremental-diffs**

Feature Information for Contextual Configuration Diff Utility

Table 3 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 3 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release. Unless noted otherwise, subsequent releases of that Cisco IOS software release also support that feature.

Table 3 Feature Information for Contextual Configuration Diff Utility

Feature Name	Releases	Feature Information
Contextual Configuration Diff Utility	12.3(4)T 12.2(25)S 12.2(27)SBC 12.2(33)SRA 12.2(33)SXH 12.2(33)SB Cisco IOS XE Release 2.1	The Contextual Configuration Diff Utility feature provides the ability to perform a line-by-line comparison of any two configuration files and generate a list of the differences between them. The generated output includes information regarding configuration lines that have been added, modified, or deleted, and the configuration modes within which a changed configuration line exists. In 12.3(4)T, this feature was introduced. In 12.2(33)SB, this feature was implemented on the Cisco 10000 series. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following sections provide information about this feature: • Benefits of the Contextual Configuration Diff Utility, page 2 • Contextual Configuration Diff Utility Output Format, page 2 • Using the Contextual Configuration Diff Utility, page 3 The following commands were modified by this feature: **show archive config differences**, **show archive config incremental-diffs**.

Loading and Maintaining System Images

Loading and Managing System Images

This chapter describes how to load and manage Cisco IOS software system images. This chapter describes tasks associated with loading microcode. System images contain the system software. Microcode typically contains system images or hardware-specific software that can be loaded directly on to various hardware devices.

For a complete description of the system image and microcode commands mentioned in this chapter, refer to the *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index, Release 12.4* or search online.

Understanding Images

Cisco IOS software is packaged in system images. Your router already has an image on it when you receive it. However, you may want to load a different image onto the router at some point. For example, you may want to upgrade your software to the latest release, or use the same version of the software for all the routers in a network. Different system images contain different sets of Cisco IOS features. To determine which version (release number) of Cisco IOS software that is running on your system, and the filename of the system image, use the **show version** command in user EXEC or privileged EXEC mode. For example, "Version 12.4" indicates Cisco IOS Release 12.4, and "c7200-js-mz" indicates the system image for a Cisco 7200 series router (c7200) containing the "enterprise" feature set (jz).

Types of Images

The following are the two main types of image your router may use:

- System image—The complete Cisco IOS software. This image is loaded when your router boots and is used most of the time.

  On most platforms, the image is located in flash memory. On platforms with multiple flash memory file systems (flash, boot flash, slot 0, slot 1, and so on), the image can be located in any existing flash file system. Use the **show file systems** privileged EXEC mode command to determine which file systems your router supports. Refer to your hardware documentation for information about where these images are located by default.

- Boot image—A subset of the Cisco IOS software. This image is used to perform network booting or to load Cisco IOS images onto the router. This image is also used if the router cannot find a valid system image. Depending on your platform, this image may be called xboot image, rxboot image, bootstrap image, or boot loader/helper image.

  On some platforms, the boot image is contained in ROM. In others, the boot image can be stored in flash memory. On these platforms, you can specify which image should be used as the boot image using the **boot bootldr** global configuration command. Refer to your hardware documentation for information about the boot image used on your router.

Image Naming Conventions

You can identify the platform, features, and image location by the name of the image. The naming convention is *platform-featureset-type* for images.

The *platform* variable indicates which platforms can use this image. Examples of *platform* variables include rsp (Cisco 7000 series with RSP7000 and Cisco 7500 series), c1600 (Cisco 1600 series), and c1005 (Cisco 1005).

The *featureset* variable identifies the feature package that the image contains. Cisco IOS software comes in feature sets tailored to suit certain operating environments, or customized for certain Cisco hardware platforms.

The *type* variable is a code indicating the characteristics of the image:

- f—The image runs from flash memory.
- m—The image runs from RAM.
- r—The image runs from ROM.
- l—The image is relocatable.
- z—The image is zip compressed.
- x—The image is mzip compressed.

General Output Conventions for Copy Operations

During a copy operation, any of the following characters may appear on the screen:

- A pound sign (#) generally means that a flash memory device is being cleared and initialized. (Different platforms use different ways of indicating that Flash is being cleared.)
- An exclamation point (!) means that ten packets have been transferred.
- A series of “V” characters means that a checksum verification of the file is occurring after the file is written to flash memory.
- An “O” means an out-of-order packet.
- A period (.) means a timeout.

The last line in the output indicates whether the copy was successful.

Working with System Images

To manage system images, perform any of the tasks in the following sections:

Displaying System Image Information

Use the following commands in privileged EXEC mode to display information about the system software:

Command	Purpose
`Router# show bootvar`	Lists the contents of the BOOT environment variable, the name of the configuration file pointed to by the CONFIG_FILE environment variable, and the contents of the BOOTLDR environment variable.
`Router# show flash-filesystem: [partition number] [all	chips	detailed	err	summary]`	Lists information about flash memory for Class B file systems.
`Router# show flash-filesystem: [all	chips	filesys]`	Lists information about flash memory for Class A file systems.
`Router# show flash-filesystem:`	Lists information about flash memory for Class C file systems.
`Router# show microcode`	Displays microcode information.
`Router# show version`	Lists the currently running system image filename, and the system software release version, the configuration register setting, and other information.

Refer to the *Cisco IOS Configuration Fundamentals Command Reference* for examples of these commands.

Copying Images from Flash Memory to a Network Server

You may want to copy image files to remote servers as a backup copy, or so that you can perform later checks by comparing the copy in flash to a saved copy.

You can copy system images from flash memory to remote servers using the FTP, the remote copy protocol (rcp), or TFTP. Cisco IOS Software Release 12.4 also supports uploading to (or downloading from) servers using HTTP or HTTPS. The following sections describe these tasks:

- Copying an Image from Flash Memory Using TFTP, page 4
- Copying an Image from Flash Memory to an rcp Server, page 5
- Copying an Image from Flash Memory to an FTP Server, page 7

The protocol you use depends on which type of server you are using. The FTP and rcp transport mechanisms provide faster performance and more reliable delivery of data than TFTP. These improvements are possible because the FTP and rcp transport mechanisms are built on and use the TCP/IP stack, which is connection-oriented.

To stop the copy process, press **Ctrl-^** or **Ctrl-Shift-6**.

In the output, an exclamation point (!) indicates that the copy process is taking place. Each exclamation point (!) indicates that ten packets have been transferred.

Refer to the *Internetwork Troubleshooting Guide* publication for procedures on how to resolve flash memory problems.

Copying an Image from Flash Memory Using TFTP

You can copy a system image to a TFTP network server. In some implementations of TFTP, you must first create a "dummy" file on the TFTP server and give it read, write, and execute permissions before copying a file over it. Refer to your TFTP documentation for more information.

To copy a system image to a TFTP network server, use the following commands in EXEC mode:

	Command	Purpose
Step 1	`Router# show flash-filesystem:`	(Optional) Displays the system image filename in Flash memory. Use this command to verify the url-path of the file and the exact spelling of the system image filename for use in the next command.
Step 2	`Router# copy flash-url tftp:[[[//location]/directory]/filename]`	Copies the system image from Flash memory to a TFTP server. Specify the file location and filename as the *flash-url* argument.

After you have issued the **copy** privileged EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying an Image from Flash Memory to a TFTP Server Example

The following example uses the **show flash:** EXEC command to learn the name of the system image file and the **copy flash: tftp:** EXEC command to copy the system image to a TFTP server:

```
RouterB# show flash:

System flash directory:
File  Length   Name/status
  1   4137888  c3640-c2is-mz.Feb24
[4137952 bytes used, 12639264 available, 16777216 total]
16384K bytes of processor board System flash (Read/Write)\

Router# copy flash: tftp:

IP address of remote host [255.255.255.255]? 172.16.13.110
```

```
filename to write on tftp host? c3640-c2is-mz.Feb24
writing c3640-c2is-mz.Feb24 !!!!...
successful tftp write.
```

Copying an Image from Partitioned Flash Memory to a TFTP Server Example

In this example, the file named your-ios is copied from partition 1 of the flash memory PC card in slot 0 to the TFTP server at 172.23.1.129. The file will be saved with the name your-ios in the dirt/sysadmin directory relative to the directory of the remote username.

```
Router# copy slot0:1:your-ios tftp://172.23.1.129/dirt/sysadmin/your-ios

Verifying checksum for 'your-ios' (file # 1)...  OK
Copy 'your-ios' from Flash to server
  as 'dirt/sysadmin/ios-2'? [yes/no] yes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Upload to server done
Flash device copy took 00:00:23 [hh:mm:ss]
```

Copying an Image from Flash Memory to an rcp Server

You can copy a system image from Flash memory to an rcp network server.

If you copy the configuration file to a PC used as a file server, the computer must support remote shell protocol (rsh).

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy an image from the router to a server using rcp, the Cisco IOS software sends the first valid username it encounters in the following list:

1. The remote username specified in the **copy** privileged EXEC command, if one if specified.
2. The username set by the **ip rcmd remote-username** global configuration command, if the command is configured.
3. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** global configuration command, the router software sends the Telnet username as the remote username.
4. The router hostname.

For the rcp copy request to execute, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written or copied relative to the directory associated with the remote username on the server. The path for all files and images to be copied begins at the remote user's home directory. For example, if the system image resides in the home directory of a user on the server, specify that user's name as the remote username.

If you are writing to the server, the rcp server must be properly configured to accept the rcp write request from the user on the router. For UNIX systems, you must add an entry to the .rhosts file for the remote user on the rcp server. For example, suppose the router contains the following configuration lines:

```
hostname Rtr1
ip rcmd remote-username User0
```

If the router's IP address translates to Router1.domain.com, then the .rhosts file for User0 on the rcp server should contain the following line:

```
Router1.domain.com Rtr1
```

Refer to the documentation for your rcp server for more information.

To copy a system image from flash memory to a rcp server, use the following commands:

	Command	Purpose
Step 1	`Router#` **`show`** *`flash-filesystem`***`:`**	(Optional) Displays the system image filename in flash memory. Use this command to verify the *url-path* of the file and the exact spelling of the system image filename for use in the **copy** privileged EXEC command.
Step 2	`Router#` **`configure terminal`**	(Optional) Enters global configuration mode from the terminal. This step is required only if you want to change the default remote username (see Step 3).
Step 3	`Router(config)#` **`ip rcmd remote-username`** *`username`*	(Optional) Configures the remote username.
Step 4	`Router(config)#` **`end`**	(Optional) Exits global configuration mode. This step is required only if you want to change the default remote username (see Step 3).
Step 5	`Router#` **`copy`** *`flash-url`* **`rcp:`**`[[[`**`//`**`[`*`username`***`@`**`]`*`location`*`]`**`/`***`directory`*`]`**`/`***`filename`*`]`	Copies the system image from flash memory to a network server using rcp.

After you have issued the **copy** privileged EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copy from Flash to RCP Server Example

The following example copies the system image named c5200-ds-l to the network server at 172.16.1.111 using rcp and a username of netadmin1:

```
Router# copy flash:c5200-ds-l rcp:netadmin1@172.16.1.111/c5200-ds-l

Verifying checksum for ‘c5200-ds-l’ (file # 1)...[OK]
Writing c5200-ds-l -
```

Copy from Slot1 to RCP Server Example

The following example copies a system image file named test from the second Personal Computer Memory Card International Association (PCMCIA) slot to a network server using rcp. The remote username is netadmin1. Because the destination address and filename are not specified, the router prompts for this information.

```
Router# configure terminal
Router(config)# ip rcmd remote-username netadmin1
Router(config)# end
Router# copy slot1:test rcp:
Address or name of remote host [UNKNOWN]? 172.16.1.111
File name to write to? test
Verifying checksum for ‘test’ (file # 1)...[OK]
Writing test
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Upload to server done
Flash device copy took 00:00:08 [hh:mm:ss]
```

Copying an Image from Flash Memory to an FTP Server

You can copy a system image from flash memory to an FTP network server.

Understanding the FTP Username and Password

The FTP protocol requires a client to send a remote username and password on each FTP request to a server. When you copy a configuration file from the router to a server using FTP, the Cisco IOS software sends the first valid username it encounters in the following list:

1. The username specified in the **copy** privileged EXEC command, if a username is specified.
2. The username set by the **ip ftp username** global configuration command, if the command is configured.
3. Anonymous.

The router sends the first valid password it encounters in the following list:

1. The password specified in the **copy** privileged EXEC command, if a password is specified.
2. The password set by the **ip ftp password** global configuration command, if the command is configured.

The router forms a password *username@routername.domain*. The variable *username* is the username associated with the current session, *routername* is the configured hostname, and *domain* is the domain of the router.

The username and password must be associated with an account on the FTP server. If you are writing to the server, the FTP server must be properly configured to accept the FTP write request from the user on the router.

If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user's name as the remote username.

Refer to the documentation for your FTP server for more information.

Use the **ip ftp username** and **ip ftp password** commands to specify a username and password for all copies. Include the username in the **copy** command if you want to specify a username for that copy operation only.

Copying from Flash Memory to an FTP Server Tasks

To copy a system image to an FTP network server, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	(Optional) Enters global configuration mode. This step is required only if you override the default remote username or password (see Steps 2 and 3).
Step 2	`Router(config)# ip ftp username username`	(Optional) Changes the default remote username.
Step 3	`Router(config)# ip ftp password password`	(Optional) Changes the default password.
Step 4	`Router(config)# end`	(Optional) Exits global configuration mode. This step is required only if you override the default remote username or password (see Steps 2 and 3).
Step 5	`Router# show flash-filesystem:`	(Optional) Displays the system image file in the specified flash directory. If you do not already know it, note the exact spelling of the system image filename in flash memory.
Step 6	`Router# copy flash-filesystem:filename ftp:[[[//[username [:password]@]location]/directory]/filename]`	Copies the image to the FTP server.

After you have issued the **copy** privileged EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying from Flash Memory to an FTP Server Example

The following example uses the **show flash: privileged** EXEC command to learn the name of the system image file and the **copy flash: tftp:** privileged EXEC command to copy the system image (c3640-c2is-mz) to a TFTP server. The router uses the default username and password.

```
Router# show flash:

System flash directory:
File  Length   Name/status
  1   4137888  c3640-c2is-mz
[4137952 bytes used, 12639264 available, 16777216 total]
16384K bytes of processor board System flash (Read/Write)\

Router# copy flash: tftp:

IP address of remote host [255.255.255.255]? 172.16.13.110
filename to write on tftp host? c3600-c2is-mz
writing c3640-c2is-mz !!!!...
successful ftp write.
```

Copying from Slot1 to an FTP Server Example

The following example uses the **show slot1: privileged** EXEC command to display the name of the system image file in the second PCMCIA slot, and copies the file (test) to an FTP server:

```
Router# show slot1:

-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time------ name
1   .. 1        46A11866 2036C   4    746      May 16 1995 16:24:37 test

Router# copy slot1:test ftp://thisuser:thatpass@172.16.13.110/test
```

```
writing test!!!!...
successful ftp write.
```

Copying from Partitioned Flash to an FTP Server Example

In this example, the file named your-ios is copied from partition 1 of the flash memory PC card in slot 0 to the TFTP server at 172.23.1.129. The file will be saved with the name your-ios in the dirt/sysadmin directory relative to the directory of the remote username.

```
Router# show slot0: partition 1

PCMCIA Slot0 flash directory, partition 1:
File  Length   Name/status
  1   1711088 your-ios
[1711152 bytes used, 2483152 available, 4194304 total]

Router# copy slot0:1:your-ios ftp://myuser:mypass@172.23.1.129/dirt/sysadmin/your-ios

Verifying checksum for 'your-ios' (file # 1)...  OK
Copy 'your-ios' from Flash to server
  as 'dirt/sysadmin/ios-2'? [yes/no] yes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Upload to server done
Flash device copy took 00:00:23 [hh:mm:ss]
```

Copying Images from a Network Server to Flash Memory

You can copy system images or boot image from a TFTP, rcp, or FTP server to a flash memory file system to upgrade or change the Cisco IOS software or boot image on your router.

The protocol you use depends on which type of server you are using. The FTP and rcp transport mechanisms provide faster performance and more reliable delivery of data than TFTP. These improvements are possible because the FTP and rcp transport mechanisms are built on and use the TCP/IP stack, which is connection-oriented.

The following sections describe the copying tasks. The first two tasks and the last task are required. If you have a run-from-flash system, the tasks in the third section are required. Perform one of the remaining tasks, depending on which file transfer protocol you use.

- Restrictions on Naming Files, page 10
- Understanding Flash Memory Space Considerations, page 10
- Output for Image Downloading Process, page 11
- Copying to Flash Memory for Run-from-Flash Systems, page 11
- Copying an Image from a TFTP Server to a Flash Memory File System, page 12
- Copying an Image from an rcp Server to a Flash Memory File System, page 14
- Copying an Image from an FTP Server to a Flash Memory File System, page 16
- Verifying the Image in Flash Memory, page 18

Note When you are upgrading or changing to a different Cisco IOS release, refer to the appropriate release notes for information on system requirements and limitations.

Restrictions on Naming Files

Filenames in flash memory can be up to 63 characters long; they are not case-sensitive and are always converted to lowercase.

Note The destination filename must be an alphanumeric expression (contains all letters or a combination of letters and numerals). For example, "1" is an invalid filename.

The filename can be in either lowercase or uppercase; the system ignores case. If more than one file of the same name is copied to flash, regardless of case, the last file copied becomes the valid file.

Understanding Flash Memory Space Considerations

Be sure that enough space is available before copying a file to flash memory. Use the **show** *flash-filesystem***:** privileged EXEC command, and compare the size of the file you want to copy to the amount of flash memory available. If the space available is less than the amount needed, the **copy** privileged EXEC command will be partially executed, but the entire file will not be copied into flash memory. The failure message "buffer overflow - *xxxx*/*xxxx*" will appear, where *xxxx*/*xxxx* is the number of bytes read from the source file and the number of bytes available on the destination device.

Caution Do not reboot the router if no valid image is in flash memory.

Note For the Cisco 3600 series routers, if you do not have access to a network server and need to download a system image, you can copy an image from a local or remote computer (such as a PC, UNIX workstation, or Macintosh) using the Xmodem or Ymodem protocol. See the section "Recovering a System Image Using Xmodem or Ymodem" later in this chapter.

On Cisco 2500, Cisco 3000, and Cisco 4000 systems, if the file being downloaded to flash memory is an uncompressed system image, the **copy** command automatically determines the size of the file being downloaded and validates it with the space available in flash memory.

On Class B flash file systems, the router gives you the option of erasing the existing contents of flash memory before writing to it. If no free flash memory is available, or if no files have ever been written to flash memory, the erase routine is required before new files can be copied. If there is enough free flash memory, the router gives you the option of erasing the existing flash memory before writing to it. The system will inform you of these conditions and prompt you for a response.

Note If you enter **n** after the "Erase flash before writing?" prompt, the copy process continues. If you enter **y** and confirm the erasure, the erase routine begins. Be sure to have ample flash memory space before entering **n** at the erasure prompt.

If you attempt to copy a file into flash memory that is already there, a prompt informs you that a file with the same name already exists. This file is deleted when you copy the new file into flash.

- On Class A and B flash file systems, the first copy of the file still resides within flash memory, but it is rendered unusable in favor of the newest version and is listed with the "deleted" tag when you use the **show** *flash-filesystem***:** privileged EXEC command. If you terminate the copy process, the newer file is marked "deleted" because the entire file was not copied and is not valid. In this case, the original file in flash memory is valid and available to the system.
- On Class C flash file systems, the first copy of the file is erased.

You can copy normal or compressed images to flash memory. You can produce a compressed system image on any UNIX platform using the **compress** interface configuration command. Refer to your UNIX platform's documentation for the exact usage of the **compress** command.

On some platforms, the flash security jumper must be installed in order to write to flash memory. In addition, some platforms have a write protect switch that must be set to *unprotected* in order to write to flash memory.

Output for Image Downloading Process

The output and dialog varies depending on the platform.

Output for Partitioned Flash Memory

One of the following prompts will be displayed after the command is entered to indicate how a file can be downloaded:

- None—The file cannot be copied.
- RXBOOT-Manual—You must manually reload to the rxboot image in ROM to copy the image.
- RXBOOT-FLH—The copy is done automatically via the flash load helper software in boot ROMs.
- Direct—The copy can be done directly.

If the file can be downloaded into more than one partition, you are prompted for the partition number. To obtain help, enter any of the following characters at the partition number prompt:

- **?**—Displays the directory listings of all partitions.
- **?1**—Displays the directory of the first partition.
- **?2**—Displays the directory of the second partition.
- **q**—Quits the **copy** command.

Copying to Flash Memory for Run-from-Flash Systems

You cannot run the system from flash memory and copy to it at the same time. Therefore, for systems that run from flash, preform either of the following tasks before copying to flash:

- Partition flash memory or use flash load helper to allow the system to run from flash memory while you copy to it.
- Reload the system to use a system image from boot ROMs.

See the "Understanding Memory Types and Functions" section in the "Maintaining System Memory" chapter of this document for more information on run-from-flash systems.

Refer to the appropriate hardware installation and maintenance publication for information about the jumper settings required for your configuration.

Copying an Image from a TFTP Server to a Flash Memory File System

Before you copy a system image or boot image to flash memory, you should make a backup copy of the current software or bootstap image. See the "Copying Images from Flash Memory to a Network Server" section on page 3 for details.

To copy a system image from a TFTP server to a flash memory file system, use the following command in EXEC mode:

Command	Purpose
`Router# copy tftp: [[[//location]/directory]/filename] flash-filesystem:[filename]`	Copies a system image or a boot image to flash memory.

After you have issued the **copy** privileged EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying from a TFTP Server to Flash Memory: Example

In the following example, a file is copied from a TFTP server to slot1:

```
Router# copy tftp://theserver/tftpboot/space2/sub2/c7200-js-mz slot1:
Destination filename [c7200-js-mz]?
Accessing tftp://theserver/tftpboot/space2/sub2/c7200-js-mz...Translating
"theserver"...domain server (192.168.2.132) [OK]

Loading tftpboot/space2/sub2/c7200-js-mz from 192.168.2.132 (via Ethernet3/0):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 4823492 bytes]

4823492 bytes copied in 264.312 secs (18270 bytes/sec)
```

The following example copies a system image named igs-p-l from a TFTP server to a Class B flash file system when flash memory is too full to copy the file:

```
Router# copy tftp: flash:

IP address or name of remote host [255.255.255.255]? dirt
Translating "DIRT"...domain server (255.255.255.255) [OK]

Name of file to copy? igs-p-l
Copy igs-p-l from 172.16.13.111 into flash memory? [confirm]
Flash is filled to capacity.
Erasure is needed before flash may be written.
Erase flash before writing? [confirm]
Erasing flash EPROMs bank 0

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
```

```
Erasing bank...eeeeeeeeeeeeeeee

Erasing flash EPROMs bank 1

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Erasing flash EPROMs bank 2

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Erasing flash EPROMs bank 3

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Loading from 172.16.1.111:!!!!...
 [OK - 1906676 bytes]
Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Flash verification successful. Length = 1906676, checksum = 0x12AD
```

Copying from a TFTP Server to Flash When a File by the Same Name Already Exists: Example

The following example shows how to copy a system image named igs-p-l into the current flash configuration in which a file named igs-p-l already exists:

```
Router# copy tftp://172.16.13.111/igs-p-l flash:igs-p-l

File igs-p-l already exists; it will be invalidated!
Copy igs-p-l from 172.16.13.111 into flash memory? [confirm]
2287500 bytes available for writing without erasure.
Erase flash before writing? [confirm]n
Loading from 172.16.1.111:!!!!...
[OK - 1906676 bytes]
Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Flash verification successful. Length = 1902192, checksum = 0x12AD
```

Copying from a TFTP Server to Flash Without a Security Jumper Installed: Example

In the following example, the flash security jumper is not installed, so you cannot write files to flash memory:

```
Router# copy tftp: flash:

Flash: embedded flash security jumper(12V)
       must be strapped to modify flash memory
```

Copying from a TFTP Server to Partitioned Flash: Example

In the following example, the file named c3600-i-mz on the TFTP server at 172.23.1.129 is copied to the first partition of internal flash Memory:

```
Router# copy tftp://172.23.1.129/c3600-i-mz flash:1:c3600-i-mz/c3600-i-mz

Accessing file 'c3600-i-mz' on 172.23.1.129...
Loading c3600-i-mz from 172.23.1.129 (via Ethernet1/0): ! [OK]
Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]
Copy 'c3600-i-mz' from server
  as 'c3600-i-mz' into Flash WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeeee ...erased
Loading c3600-i-mz from 172.23.1.129 (via Ethernet1/0):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 1711088 bytes]

Verifying checksum...  OK (0xF89A)
Flash device copy took 00:00:17 [hh:mm:ss]
```

Copying an Image from an rcp Server to a Flash Memory File System

You can copy a system image from an rcp network server to a flash memory file system.

If you copy the configuration file to a PC used as a file server, the computer must support rsh.

Understanding the rcp Username

The rcp protocol requires a client to send a remote username on each rcp request to a server. When you copy an image from the router to a server using rcp, the Cisco IOS software sends the first valid username it encounters in the following list:

1. The remote username specified in the **copy** privileged EXEC command, if one if specified.
2. The username set by the **ip rcmd remote-username** global configuration command, if the command is configured.
3. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the router through Telnet and was authenticated through the **username** global configuration command, the router software sends the Telnet username as the remote username.
4. The router hostname.

For the rcp copy request to execute, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written or copied relative to the directory associated with the remote username on the server. The path for all files and images to be copied begins at the remote user’s home directory. For example, if the system image resides in the home directory of a user on the server, specify that user’s name as the remote username.

Copying from an rcp Server to Flash Memory

To copy an image from an rcp server to flash memory, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`See the instructions in the section "Copying Images from Flash Memory to a Network Server."`	Make a backup copy of the current system or bootstrap software image.
Step 2	`Router#` **`configure terminal`**	(Optional) Enters global configuration mode from the terminal. This step is required only if you override the default remote username (see Step 3).
Step 3	`Router(config)#` **`ip rcmd remote-username`** *`username`*	(Optional) Specifies the remote username.
Step 4	`Router#` **`end`**	(Optional) Exits global configuration mode. This step is required only if you override the default remote username (see Step 3).
Step 5	`Router#` **`copy rcp:`** `[[[//[`*`username`*`@]`*`location`*`]/`*`directory`*`] /`*`filename`*`]` *`flash-filesystem`*`:[`*`filename`*`]`	Copies the image from an rcp server to a Flash memory file system.

After you have issued the **copy** privileged EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying from an rcp Server to Flash Example

The following example copies a system image named mysysim1 from the netadmin1 directory on the remote server named SERVER1.CISCO.COM with an IP address of 172.16.101.101 to flash memory. To ensure that enough flash memory is available to accommodate the system image to be copied, the Cisco IOS software allows you to first erase the contents of flash memory.

```
Router1# configure terminal
Router1(config)# ip rcmd remote-username netadmin1
Router1(config)# end
Router# copy rcp: flash:

System flash directory:
File name/status
   1 mysysim1
[2076072 bytes used, 21080 bytes available]

Address or name of remote host[UNKNOWN]? 172.16.101.101
Name of file to copy? mysysim1
Copy mysysim1 from SERVER1.CISCO.COM?[confirm]

Checking for file 'mysysim1' on SERVER1.CISCO.COM...[OK]

Erase Flash device before writing?[confirm]
Are you sure?[confirm]
Erasing device...ezeeze...erased.

Connected to 172.16.101.101

Loading 2076007 byte file mysysim1:!!!!...
[OK]

Verifying checksum... (0x87FD)...[OK]
```

Copying from an rcp Server to Partitioned Slot0: Example

In the following example, the file named c3600-i-mz on the rcp server at the IP address 172.23.1.129 is copied to partition 3 in slot 0. Because no username is specified, the router uses the default rcp remote username.

```
Router# show slot0: partition 3

PCMCIA Slot0 flash directory, partition 3:
File  Length   Name/status
  1   426      running-config
[492 bytes used, 4193812 available, 4194304 total]

Router# copy rcp://172.23.1.129/tftpboot/gate/c3600-i-mz slot0:3:/tftpboot/gate/c3600-i-mz

Accessing file '/tftpboot/gate/c3600-i-mz' on 172.23.1.129...
Connected to 172.23.1.129
Loading 1711088 byte file c3600-i-mz: ! [OK]

Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]
Copy '/tftpboot/gate/c3600-i-mz' from server
  as '/tftpboot/gate/c3600-i-mz' into Flash WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
Connected to 172.23.1.129
Loading 1711088 byte file c3600-i-mz:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! [OK]

Verifying checksum...  OK (0xF89A)
Flash device copy took 00:00:16 [hh:mm:ss]
```

Copying an Image from an FTP Server to a Flash Memory File System

You can copy a system image from an FTP server to a flash memory file system.

Understanding the FTP Username and Password

The FTP protocol requires a client to send a remote username and password on each FTP request to a server. When you copy a configuration file from the router to a server using FTP, the Cisco IOS software sends the first valid username it encounters in the following list:

1. The username specified in the **copy** privileged EXEC command, if a username is specified.
2. The username set by the **ip ftp username** global configuration command, if the command is configured.
3. Anonymous.

The router sends the first valid password it encounters in the following list:

1. The password specified in the **copy** privileged EXEC command, if a password is specified.
2. The password set by the **ip ftp password** command, if the command is configured.

The router forms a password *username@routername.domain*. The variable *username* is the username associated with the current session, *routername* is the configured host name, and *domain* is the domain of the router.

The username and password must be associated with an account on the FTP server. If you are writing to the server, the FTP server must be properly configured to accept the FTP write request from the user on the router.

If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user's name as the remote username.

Refer to the documentation for your FTP server for more information.

Use the **ip ftp username** and **ip ftp password** commands to specify a username and password for all copies. Include the username in the **copy** command if you want to specify a username for that copy operation only.

Copying from an FTP Server to Flash Memory

To copy a system image from an FTP server to a flash memory file system, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	See the instructions in the section "Copying Images from Flash Memory to a Network Server."	Make a backup copy of the current software image or bootstrap image.
Step 2	`Router#` **`configure terminal`**	(Optional) Enters global configuration mode from the terminal. This step is required only if you want to override the default remote username or password (see Steps 3 and 4).
Step 3	`Router(config)#` **`ip ftp username`** *`username`*	(Optional) Changes the default remote username.
Step 4	`Router(config)#` **`ip ftp password`** *`password`*	(Optional) Changes the default password.
Step 5	`Router(config)#` **`end`**	(Optional) Exits global configuration mode. This step is required only if you override the default remote username or password (see Steps 3 and 4).
Step 6	`Router#` **`copy ftp:`** `[[[//[`*`username`*`[:`*`password`*`]@]`*`location`*`]/`*`directory`*`]/`*`filename`*`]` *`flash-filesystem`*`:[`*`filename`*`]`	Copies the configuration file from a network server to running memory or the startup configuration using rcp.

After you have issued the **copy** privileged EXEC command, you may be prompted for additional information or for confirmation of the action. The prompting will depend on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copy from FTP Server to Flash Memory Example

The following example copies a the file named c7200-js-mz from the FTP server the server using a username of myuser and a password of mypass:

```
Router# copy ftp://myuser:mypass@theserver/tftpboot/sub3/c7200-js-mz slot1:c7200-js-mz

Accessing ftp://theserver/tftpboot/sub3/c7200-js-mz...Translating "theserver"...domain
server (192.168.2.132) [OK]

Loading c7200-js-mz from 192.168.2.132 (via Ethernet3/0):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 4823492 bytes]

4823492 bytes copied in 264.312 secs (18270 bytes/sec)
```

Verifying the Image in Flash Memory

Before booting from flash memory, use the **verify** privileged EXEC command to verify that the checksum of the image in flash memory matches the checksum listed in the README file that was distributed with the system software image. The checksum of the image in flash memory is displayed at the bottom of the screen when you issue the **copy** privileged EXEC command to copy an image. The README file was copied to the network server automatically when you installed the system software image on the server.

Caution If the checksum value does not match the value in the README file, do not reboot the router. Instead, issue the **copy** command and compare the checksums again. If the checksum repeatedly is incorrect, copy the original system software image back into flash memory *before* you reboot the router from flash memory. If you have a corrupted image in flash memory and try to boot from flash, the router will start the system image contained in ROM (assuming that booting from a network server is not configured). If ROM does not contain a fully functional system image, the router will not function and must be reconfigured through a direct console port connection.

The flash memory content listing does not include the checksum of individual files. To recompute and verify the image checksum after an image is copied into flash memory or a flash memory device, use the following command in privileged EXEC mode:

Command	Purpose
`Router#` **`verify`** *`flash-filesystem`***`:`**`[`*`partition-number`***`:`**`] [`*`filename`*`]`	Recomputes and verifies the image checksum after the image is copied into flash memory.

If you do not provide the filename in the command, the router prompts you. By default, it prompts for the last (most recent) file in flash. Press Return to recompute the default file checksum, or enter the filename of a different file at the prompt. Note that the checksum for microcode images is always 0x0000.

The following example verifies the image named c7200-js-mz in slot0:

```
Router# verify slot0:c7200-js-mz

Verified slot0:c7200-js-mz
```

Copying Images Using HTTP or HTTPS

Cisco IOS Release 12.4 supports file transfers between your Cisco IOS software-based device and a remote HTTP server using the HTTP or Secure HTTP (HTTPS) protocol.

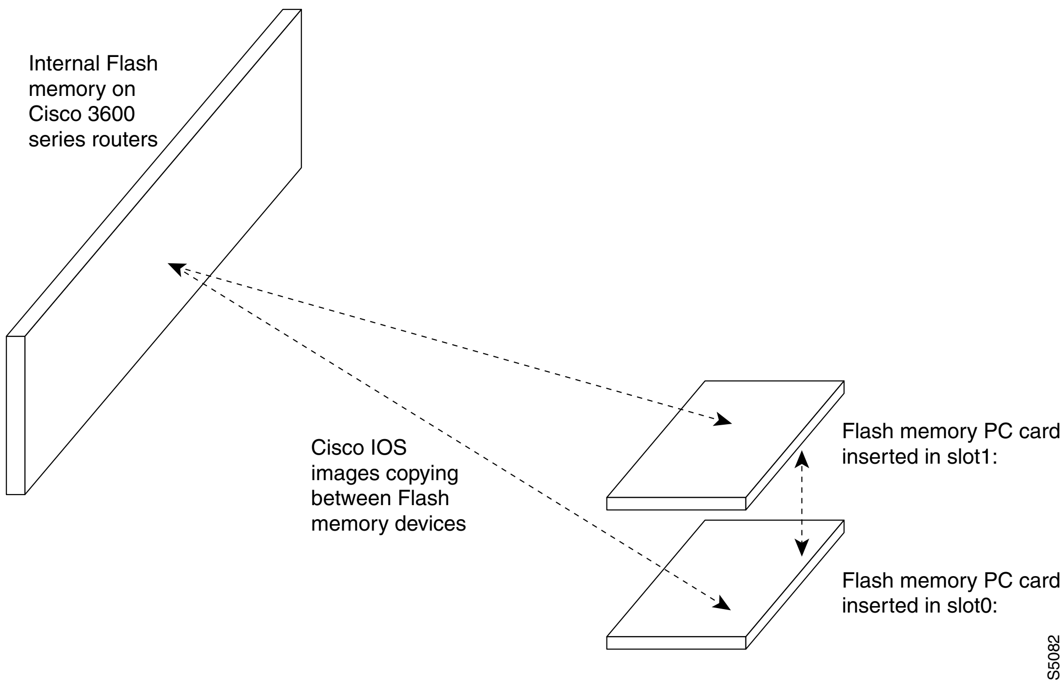
To copy files to or from a remote HTTP server, your system image must support the HTTP Client feature, which is integrated in most Cisco IOS software images. The HTTP client is enabled by default. To determine if the HTTP client is supported on your system, issue the **show ip http client all** privileged EXEC mode command. If you are able to execute the command, the HTTP client is supported.

For a complete description of this feature, see the "Transferring Files Using HTTP or HTTPS" module.

Copying Images Between Local Flash Memory Devices

On routers with multiple flash memory devices, you can copy images from one flash memory file system, such as internal flash memory or a flash memory card in a PCMCIA slot, to another flash memory device, as shown in Figure 9. One reason to copy the image to a different flash device is to make a backup copy of it.

Figure 9 Copying Images Between Flash Memory File Systems



Caution Before copying to a new flash device, you must first format that device.
All new media should be formatted. Memory media used in Cisco devices does not typically come preformatted. Even if preformatted, an initial format using the Cisco files system may help to prevent potential problems with incompatible formatting.
Attempts to copy images to unformatted or improperly formatted flash devices may not generate failure messages on some devices. For this reason, the **show** and **verify** steps shown in the following table are strongly recommended.
For instructions on formatting your flash device, see the "Maintaining System Memory" chapter.

To copy an image between flash memory devices, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# show flash-filesystem:`	Displays the layout and contents of flash memory.
Step 2	`Router# copy source-url destination-url`	Copies an image between flash memory devices.
Step 3	`Router# verify flash-filesystem:filename`	Verifies the checksum of the image you copied. (You can get the MD5 checksum for your image from Cisco.com).

Note The source device and the destination device cannot be the same. For example, the **copy slot1: slot1:** command is invalid.

Copying a File Between Local Flash Memory Devices Example

The following example copies the file named new-ios from partition 1 of internal flash memory to slot 0:

```
Router# show flash: partition 1

System flash directory, partition 1:
File  Length   Name/status
  1   3142748 admin/images/new-ios
[3142812 bytes used, 1051492 available, 4194304 total]

Router# show slot0:

PCMCIA Slot0 flash directory
File  Length   Name/status
  1   1711088  /tftpboot/gate/c3600-i-mz
[1711152 bytes used, 2483152 available, 4194304 total]

Router# copy flash:1:admin/images/new-ios slot0:admin/images/new-ios

Verifying checksum for 'admin/images/new-ios' (file # 1)...  OK

Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]
Copy 'admin/images/new-ios' from flash: device
  as 'admin/images/new-ios' into slot0: device WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 [OK - 3142748 bytes]

Flash device copy took 00:00:50 [hh:mm:ss]
Verifying checksum...  OK (0xB732)

Router# show slot0:
```

```
PCMCIA Slot0 flash directory
File  Length   Name/status
  1   3142748 admin/images/new-ios
[3142812 bytes used, 1051492 available, 4194304 total]

Router# verify slot0:
Verify filename []? new-ios
! long pause ...
Verifying file integrity of slot0:new-ios....................!
Embedded Hash   MD5 : E1A04D4DE1ED00407E6E560B315DA505
Computed Hash   MD5 : E1A04D4DE1ED00407E6E560B315DA505
CCO Hash        MD5 : C03EC4564F86F9A24201C88A9DA67317

Signature Verified
Verified slot0:

Router#
```

Specifying the Startup System Image in the Configuration File

You can enter multiple boot commands in the startup configuration file or in the BOOT environment variable to provide backup methods for loading a system image onto the router. The following are three ways to load a system image:

- From flash memory—Flash memory allows you to copy new system images without changing ROM. Information stored in flash memory is not vulnerable to network failures that might occur when loading system images from servers.
- From a network server—In case flash memory becomes corrupted, you can specify that a system image be loaded from a network server using Maintenance Operation Protocol (MOP), TFTP, rcp, or FTP as a backup boot method. For some platforms, you can specify a boot image to be loaded from a network server using TFTP, rcp, or FTP.
- From ROM—In case of both flash memory corruption and network failure, specifying a system image to be loaded from ROM provides a final backup boot method. System images stored in ROM may not always be as current as those stored in flash memory or on network servers.

Note Some platforms cannot boot from ROM.

You can enter the different types of boot commands in any order in the startup configuration file or in the BOOT environment variable. If you enter multiple boot commands, the Cisco IOS software tries them in the order they are entered.

Note Booting from ROM is faster than booting from flash memory. However, booting from flash memory is faster and more reliable than booting from a network server.

Loading the System Image from Flash Memory

Use the tasks described in the following sections to configure your router to boot from flash memory. Flash memory can reduce the effects of network failure by reducing dependency on files that can be accessed only over the network.

Configuring Flash Memory

To configure the router to load a system image in flash memory, perform the following steps:

	Task
Step 1	(Optional) Copy a system image or boot image to flash memory using TFTP, rcp, or FTP. See the "Copying Images from a Network Server to Flash Memory" section for more information on performing this step.
Step 2	Configure the system to automatically boot from the desired file and location in flash memory or boot flash memory. See the "Configuring the Router to Automatically Boot from an Image in Flash Memory" section.
Step 3	(Optional) Depending on the current configuration register setting, change the configuration register value. See the "Configuring the Router to Automatically Boot from an Image in Flash Memory" section for more information on modifying the configuration register.
Step 4	(Optional) For some platforms, set the BOOTLDR environment variable to change the location of the boot image.
Step 5	Save your configuration.
Step 6	Power-cycle and reboot your system to ensure that all is working as expected.

Configuring the Router to Automatically Boot from an Image in Flash Memory

To configure a router to automatically boot from an image in flash memory, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode from the terminal.
Step 2	`Router(config)# boot system flash [flash-filesystem:] [partition-number:] filename`	Specifies the filename of an image stored in flash memory that should be used for booting.
Step 3	`Router(config)# config-register value`	Sets the configuration register to enable loading of the system image specified in the configuration file.
Step 4	`Router(config)# end`	Ends your configuration session and exits global configuration mode.
Step 5	`Router# copy system:running-config nvram:startup-config`	Saves the system running configurattion as the device startup configuration (startup-config file).
Step 6	`Router# more nvram:startup-config`	(Optional) Allows verification of the contents of the startup configuration.
Step 7	`Router# reload`	Reboots the system.

For routers that are partitioned, if you do not specify a partition, the router boots from the first partition. If you do not specify a filename, the router boots from the first valid image found in the partition.

If you enter more than one image filename, the router tries the filenames in the order entered.

To remove a filename from the configuration file, enter the **no boot system flash** global configuration command and specify the file location.

Note The **no boot system** configuration command disables all **boot system** configuration commands regardless of argument. Specifying the **flash** keyword or the *filename* argument with the **no boot system** command disables only the commands specified by these arguments.

Configuring the Router to Boot from Flash Memory Example

The following example shows a router configured to automatically boot from an image in flash memory:

```
Router# configure terminal
Router(config)# boot system flash new-image
Router(config)# config-register 0x010F
Router(config)# end
Router# copy system:running-config nvram:startup-config
[ok]
Router# reload
[confirm]

%SYS-5-RELOAD: Reload requested
System Bootstrap, Version 12.0(7)W5(15) RELEASE SOFTWARE
     Copyright (c) 1986-2001 by Cisco Systems, Inc.
RP1 processor with 16384 Kbytes of memory
F3: 1871404+45476+167028 at 0x1000

Booting new-image from flash memory RRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR [OK - 1916912 bytes]
F3: 1871404+45476+167028 at 0x1000

              Restricted Rights Legend
.
.
.
```

Loading the System Image from a Network Server

You can configure the Cisco IOS software to load a system image from a network server using FTP, TFTP, rcp, or MOP.

If you do not boot from a network server using MOP and you do not specify either FTP, TFTP, or rcp, by default the system image that you specify is booted from a network server via TFTP.

Note If you are using a Sun workstation as a network server and TFTP to transfer the file, configure the workstation to enable verification and generation of User Datagram Protocol (UDP) checksums. See Sun documentation for details.

For increased performance and reliability, use rcp to boot a system image from a network server. The rcp implementation uses TCP, which ensures reliable delivery of data.

You cannot explicitly specify a remote username when you issue the **boot** ROM monitor command. Instead, the hostname of the router is used. If the remote server has a directory structure, as do UNIX systems, and you boot the router from a network server using rcp, the Cisco IOS software searches for the system image on the server relative to the directory of the remote username.

You can also boot from a compressed image on a network server. One reason to use a compressed image is to ensure that enough memory is available for storage. On routers that do not contain a run-from-ROM image in EPROM, when the router boots software from a network server, the image being booted and the running image both must fit into memory. If the running image is large, there may not be room in memory for the image being booted from the network server.

If not enough room is in memory to boot a regular image from a network server, you can produce a compressed software image on any UNIX platform using the **compress** interface configuration command. Refer to your UNIX platform's documentation for more information on using of the **compress** command.

To specify the loading of a system image from a network server, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# boot system [rcp | tftp] filename [ip-address]` or `Router(config)# boot system mop filename [mac-address] [interface]`	Specifies the system image file to be booted from a network server using rcp, TFTP, or MOP.
Step 3	`Router(config)# config-register value`	Sets the configuration register to enable loading of the image specified in the configuration file.
Step 4	`Router(config)# exit`	Exits configuration mode.
Step 5	`Router# copy system:running-config nvram:startup-config` or `Router# copy run start`	Saves the configuration file to your startup configuration.

In the following example, a router uses rcp to boot from the testme5.tester system image file on a network server at IP address 172.16.0.1:

```
Router# configure terminal
Router(config)# boot system rcp testme5.tester 172.16.0.1
Router(config)# config-register 0x010F
Router(config)# exit
Router# copy system:running-config nvram:startup-config
```

The following section describes how to change request retry times and frequency if you have configured your system to boot using the **boot system mop** command.

Changing MOP Request Parameters

If you configure your router to boot from a network server using MOP (using the **boot system mop** global configuration mode command), the router will send a request for the configuration file to the MOP boot server during startup. By default, when the software sends a request that requires a response from a MOP boot server and the server does not respond, the message will be re-sent after 4 seconds. The message will be re-sent a maximum of eight times. The MOP device code is set to the Cisco device code by default.

If the MOP boot server and router are separated by a slow serial link, it may take longer than 4 seconds for the router to receive a response to its message. Therefore, you may want to configure the software to wait longer than 4 seconds before resending the message if you are using such a link. You may also want to change the maximum number of retries for the MOP request or the MOP device code.

To change the Cisco IOS software request parameters for sending boot requests to a MOP server, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode from the terminal.
Step 2	`Router(config)#` **`mop device-code`** `{`**`cisco`** `|` **`ds200`**`}` **`mop retransmit-timer`** *`seconds`* **`mop retries`** *`count`*	Changes MOP server parameters.
Step 3	`Router(config)#` **`end`**	Exits global configuration mode.
Step 4	`Router#` **`copy running-config startup-config`**	Saves the configuration file to your startup configuration.

In the following example, if the MOP boot server does not respond within 10 seconds after the router sends a message, the software will resend the message:

```
Router# configure terminal
Router (config)# mop retransmit-timer 10
Router (config)# end
Router# copy running-config startup-config
```

Loading the System Image from ROM

To load the ROM system image as a backup to other boot instructions in the configuration file, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`boot system rom`**	Specifies use of the ROM system image as a backup image.
Step 3	`Router(config)#` **`config-register`** *`value`*	Sets the configuration register to enable loading of the system image specified in the configuration file.
Step 4	`Router(config)#` **`end`**	Exits global configuration mode.
Step 5	`Router#` **`copy system:running-config nvram:startup-config`**	Saves the configuration file to your startup configuration.

In the following example, a router is configured to boot from ROM:

```
Router# configure terminal
Router(config)# boot system rom
Router(config)# config-register 0x010F
Router(config)# end
```

```
Router# copy system:running-config nvram:startup-config
```

Note The Cisco 7000 series routers cannot load from ROM.

Using a Fault-Tolerant Booting Strategy

Occasionally network failures make booting from a network server impossible. To lessen the effects of network failure, consider the following booting strategy. After flash is installed and configured, you may want to configure the router to boot in the following order:

1. Boot an image from flash.
2. Boot an image from a network server.
3. Boot from ROM image.

This boot order provides the most fault-tolerant booting strategy. Use the following commands beginning in privileged EXEC mode to allow the router to boot first from flash, then from a system file from a network server, and finally from ROM:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# boot system flash [flash-filesystem:][partition-number:] filename`	Configures the router to boot from flash memory.
Step 3	`Router(config)# boot system [rcp | tftp] filename [ip-address]`	Configures the router to boot from a network server.
Step 4	`Router(config)# boot system rom`	Configures the router to boot from ROM.
Step 5	`Router(config)# config-register value`	Sets the configuration register to enable loading of the system image specified in the configuration file.
Step 6	`Router(config)# end`	Exits global configuration mode.
Step 7	`Router# copy system:running-config nvram:startup-config`	Saves the configuration file to your startup configuration.

In the following example, a router is configured to first boot an internal flash image named *gsxx*. Should that image fail, the router will boot the configuration file *gsxx* from a network server. If that method should fail, then the system will boot from ROM.

```
Router# configure terminal
Router(config)# boot system flash gsxx
Router(config)# boot system gsxx 172.16.101.101
Router(config)# boot system rom
Router(config)# config-register 0x010F
Router(config)# end
Router# copy system:running-config nvram:startup-config
[ok]
```

Using this strategy, a router has three alternative sources from which to boot. These alternative sources help lessen the negative effects of a failure on network or file server.

Recovering a System Image Using Xmodem or Ymodem

If you do not have access to a network server and need to download a system image (to update it, or if all the system images in flash memory somehow are damaged or erased), you can copy an image from a local or remote computer (such as a PC, UNIX workstation, or Macintosh) using the Xmodem or Ymodem protocol. This functionality primarily serves as a disaster recovery technique and is illustrated in Figure 10.

Note Recovering system images using Xmodem or Ymodem is performed only on the Cisco 1600 series and Cisco 3600 series routers.

Xmodem and Ymodem are common protocols used for transferring files and are included in applications such as Windows 3.1 (TERMINAL.EXE), Windows 95 (HyperTerminal), Windows NT 3.5x (TERMINAL.EXE), Windows NT 4.0 (HyperTerminal), and Linux UNIX freeware (minicom).

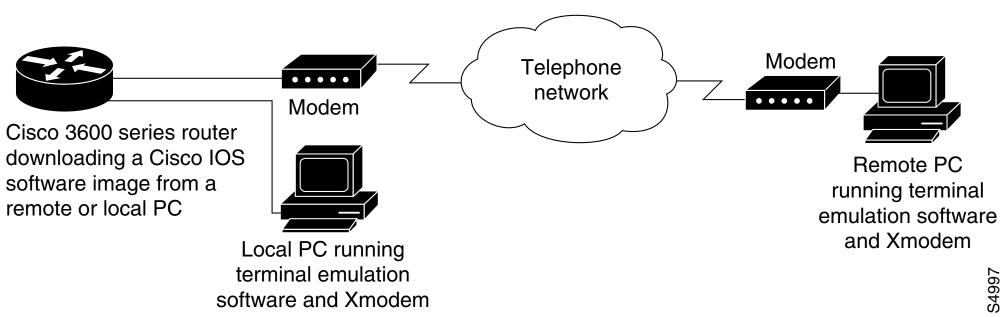
Cisco 3600 series routers do not support XBOOT functionality, a disaster recovery technique for Cisco IOS software, and do not have a separate boot helper (rxboot) image.

Xmodem and Ymodem downloads are slow, so you should use them only when you do not have access to a network server. You can speed up the transfer by setting the transfer port speed to 115200 bps.

On the Cisco 3600 series routers, you can perform the file transfer using Cisco IOS software or, if all local system images are damaged or erased, the ROM monitor. When you use Cisco IOS software for an Xmodem or Ymodem file transfer, the transfer can occur on either the AUX port or the console port. We recommend the AUX port, which supports hardware flow control. File transfers from the ROM monitor must use the console port.

On the Cisco 1600 series routers, you can perform the file transfer only from the ROM monitor over the console port.

Figure 10 Copying a System Image to a Cisco 3600 Series Router with Xmodem or Ymodem



To copy a Cisco IOS image from a computer or workstation to a router using the Xmodem or Ymodem protocol, use the following commands, as needed:

	Command	Purpose
Step 1	Router# **copy xmodem:** *flash-filesystem*:[*partition*:][*filename*] or Router# **copy ymodem:** *flash-filesystem*:[*partition*:][*filename*]	Copies a system image from a computer to flash memory using Cisco IOS software in EXEC mode (Cisco 3600 series routers only).
Step 2	ROMMON > **xmodem** [**-c**] [**-y**] [**-e**] [**-f**] [**-r**] [**-x**] [**-s** *data-rate*] [*filename*]	Copies a system image from a computer to flash memory in ROM monitor mode for the Cisco 1600 series routers. The **-c** option provides CRC-16 checksumming; **-y** uses the Ymodem protocol; **-e** erases the first partition in flash memory; **-f** erases all of flash memory; **-r** downloads the image to DRAM (the default is flash memory); **-x** prevents the image from executing after download; and **-s** sets the console port data rate.
Step 3	ROMMON > **xmodem** [**-c** | **-y** | **-r** | **-x**] [*filename*]	Copies a system image from a computer to flash memory in ROM monitor mode for the Cisco 3600 series routers.

The computer from which you transfer the Cisco IOS image must be running terminal emulation software and the Xmodem or Ymodem protocol.

For the Cisco 1600 series routers, if you include the **-r** option (download to DRAM), your router must have enough DRAM to hold the file being transferred. To run from flash memory, an image must be positioned as the first file in flash memory. If you are copying a new image to boot from flash memory, erase all existing files first.

Xmodem Transfer Using the Cisco IOS Software

The following task shows a file transfer using Cisco IOS software and the Xmodem protocol. The Ymodem protocol follows a similar procedure, using the **copy ymodem:** privileged EXEC command.

Note This functionality is enabled on Cisco 3600 series routers only.

To transfer a Cisco IOS image from a computer running terminal emulation software and the Xmodem protocol, perform the following steps:

Step 1 Place a Cisco IOS software image on the remote computer's hard drive. You can download an image from Cisco.com.

Step 2 To transfer from a remote computer, connect a modem to the AUX port of your Cisco 3600 series router and to the standard telephone network. The AUX port is set by default to a speed of 9600 bps, 2 stop bits, and no parity. The maximum speed is 115200 bps. Configure the router for both incoming and outgoing calls by entering the **modem inout** line configuration command.

Connect a modem to the remote computer and to the telephone network. The remote computer dials through the telephone network and connects to the router.

To transfer from a local computer, connect the router's AUX port to a serial port on the computer, using a null-modem cable. The AUX speed configured on the router must match the transfer speed configured on the local computer.

Step 3 At the privileged EXEC prompt in the terminal emulator window of the computer, enter the **copy xmodem: flash:** privileged EXEC command:

```
Router# copy xmodem: flash:
                        **** WARNING ****
x/ymodem is a slow transfer protocol limited to the current speed
settings of the auxiliary/console ports. The use of the auxiliary
port for this download is strongly recommended.
During the course of the download no exec input/output will be
available.
                        ---- ******* ----
```

Step 4 Press **Enter** to continue.

Step 5 Specify whether to use cyclic redundancy check (CRC) block checksumming, which verifies that your data has been correctly transferred from the computer to the router. If your computer does not support CRC block checksumming, enter **no** at the prompt:

```
Proceed? [confirm]
Use crc block checksumming? [confirm] no
```

Step 6 Determine how many times the software should try to receive a bad block of data before it declares the copy operation a failure. The default is ten retries. A higher number may be needed for noisy telephone lines. You can configure an unlimited number of retries.

```
Max Retry Count [10]: 7
```

Step 7 Decide whether you want to check that the file is a valid Cisco 3600 series image:

```
Perform image validation checks? [confirm]
Xmodem download using simple checksumming with image validation
Continue? [confirm]
```

After the transfer has begun, and if the image is valid, the software determines whether enough flash memory space exists on the router to accommodate the transfer:

```
System flash directory:
File  Length   Name/status
  1   1738244  images/c3600-i-mz
[1738308 bytes used, 2455996 available, 4194304 total]
```

Step 8 Enter the destination filename:

```
Destination file name ? new-ios-image
```

Step 9 If you do not want the contents of internal flash memory erased before the file transfer, enter **no**:

```
Erase flash device before writing? [confirm] no

Copy '' from server
  as 'new-ios-image' into Flash WITHOUT erase? [yes/no] yes
Ready to receive file...........
```

Step 10 Start an Xmodem or Ymodem send operation with the terminal emulation software on the computer that is sending the system image to the router. See your emulation software application's documentation for instructions on how to execute a file transfer. Depending on the application you use, the emulation software may display the progress of the file transfer.

Xmodem Transfer Using the ROM Monitor

This task shows a file transfer using the ROM monitor and the Xmodem protocol. To send with the Ymodem protocol, use the **xmodem -y** ROM monitor command.

For the Cisco 3600 series routers, the router must have enough DRAM to hold the file being transferred, even if you are copying to flash memory. The image is copied to the first file in internal flash memory. Any existing files in flash memory are erased. Copying files to flash partitions or to the second-file position is not supported.

Caution A modem connection from the telephone network to your console port introduces security issues that you should consider before enabling the connection. For example, remote users can dial in to your modem and access the router's configuration settings.

Step 1 Place a Cisco IOS software image on the remote computer's hard drive. You can download an image from Cisco.com or from the Feature Pack (Cisco 1600 series routers only).

Step 2 To transfer from a remote computer, connect a modem to the console port of your router and to the standard telephone network. The modem and console port must communicate at the same speed, which can be from 9600 to 115200 bps (Cisco 3600 series routers) or from 1200 to 115200 bps (Cisco 1600 series routers), depending on the speed supported by your modem. Use the **confreg** ROM monitor command to configure the console port transmission speed for the router. For the Cisco 1600 series routers, you can also set the transmission speed with the **-s** option.

Connect a modem to the remote computer and to the telephone network. The remote computer dials through the telephone network and connects to the router.

To transfer from a local computer, connect the router's console port to a serial port on the computer, using a null-modem cable. The console port speed configured on the router must match the transfer speed configured on the local computer.

Note If you are transferring from a local computer, you may need to configure the terminal emulation program to ignore Request To Send (RTS)/data terminal ready (DTR) signals.

Step 3 You should see a ROM monitor prompt in the terminal emulation window:

```
rommon >
```

Enter the **xmodem** ROM monitor command, along with any desired copy options and, optionally, the filename of the Cisco IOS image. The image loads into flash memory by default; to download to DRAM instead, use the **-r** option. The image is normally executed on completion of the file transfer; to prevent execution, use the **-x** option. The **-c** option specifies CRC-16 checksumming, which is more sophisticated and thorough than standard checksumming, if it is supported by the computer:

```
rommon > xmodem -c new-ios-image

Do not start the sending program yet...
         File size           Checksum   File name
   1738244 bytes (0x1a8604)   0xdd25 george-admin/c3600-i-mz


WARNING: All existing data in flash will be lost!
Invoke this application only for disaster recovery.
Do you wish to continue? y/n  [n]:  yes
Ready to receive file new-ios-image ...
```

Step 4 Start an Xmodem send operation, which is initiated from the terminal emulation software on the remote computer that is sending the system image to the router. See your emulation software application's documentation for instructions on how to execute an Xmodem file transfer.

Step 5 The Cisco IOS image is transferred and executed. If you are transferring from a remote computer, the computer maintains control of your console port even after the new Cisco IOS image is running. To release control to a local terminal, reconfigure the speed of the router's console port to match the speed of the local terminal by entering the **speed** *bps* line configuration command from the remote computer at the router prompt:

```
Router# configure terminal
Router(config)# line 0
Router(config-line)# speed 9600
```

The remote connection is broken, and you can disconnect the modem from the console port and reconnect the terminal line.

Loading, Upgrading, and Verifying Microcode Images

On some Cisco routers, including Cisco 7200, 7500, and 12000 series Internet routers, you can update microcode by loading it into peripheral components. This section provides information on loading, upgrading, and verifying microcode images, as described in the following subsections:

- Understanding Microcode Images, page 31
- Specifying the Location of the Microcode Images, page 32
- Reloading the Microcode Image, page 32
- Displaying Microcode Image Information, page 33

Understanding Microcode Images

Microcode is stored on ROM and allows the addition of new machine instructions without requiring that they be designed into electronic circuits when new instructions are needed. Microcode images contain microcode software that runs on various hardware devices. For example, microcode can be updated in Channel Interface Processors (CIPs) on Cisco 7500 series routers, or in Channel Port Adapters (CPAs) on Cisco 7200 series routers.

By default, the system loads the microcode bundled with the Cisco IOS system software image. This microcode is referred to as the default microcode image. However, you can configure the router to use microcode stored in flash.

Cisco 7000 series routers with an RSP7000 and Cisco 7500 series routers each have a writable control store (WCS) that stores microcode. You can load updated microcode onto the WCS from boot flash or from a flash memory card inserted in one of the PCMCIA slots of the Route/Switch Processor (RSP) card.

You can update microcode without having physical access to the router by using the **copy** privileged EXEC command to copy microcode to a flash file system.

Specifying the Location of the Microcode Images

To specify the location from where the microcode image should be loaded, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`copy tftp: flash:`** or `Router#` **`copy tftp:`** *`file-id`*	(Optional) Copies microcode files into flash. Perform this step only if you want to load the microcode from flash. See the section "Copying Images from a Network Server to Flash Memory" for more information about how to copy images to flash memory.
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`microcode`** *`interface`* `[`*`flash-filesystem`***`:`***`filename`* `[`*`slot`*`] |` **`system`** `[`*`slot`*`]]`	Configures the router to load microcode on a target interface from the specified memory location.
Step 4	`Router(config)#` **`end`**	Exits global configuration mode.
Step 5	`Router#` **`copy system:running-config nvram:startup-config`**	Saves the new configuration information.

If an error occurs when you are attempting to download a microcode image, the system loads the default system microcode image.

Note Microcode images cannot be compressed.

Reloading the Microcode Image

The configuration commands specifying the microcode to load are implemented following one of three events:

- The system is booted.
- A card is inserted or removed.
- The **microcode reload** global configuration command is issued.

After you have entered a microcode configuration command and one of these events has taken place, all cards are reset, loaded with microcode from the appropriate sources, tested, and enabled for operation.

To signal to the system that all microcode configuration commands have been entered and the processor cards should be reloaded, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`microcode reload`**	Reloads the microcode from the source specified in the configuration on to all interface and processor cards.

Immediately after you enter the **microcode reload** global configuration command and press Return, the system reloads all microcode. Global configuration mode remains enabled. After the reload is complete, enter the **exit** global configuration command to return to the privileged EXEC prompt.

If flash memory is busy because a card is being removed or inserted, or a **microcode reload** command is executed while flash is locked, the files will not be available and the onboard ROM microcode will be loaded. Issue another **microcode reload** command when flash memory is available, and the proper microcode will be loaded. The **show flash** privileged EXEC command will reveal if another user or process has locked flash memory.

Note The **microcode reload** command should not be used while flash is in use. For example, do not use this command when a **copy** {**ftp:** | **rcp:** | **tftp:**} *flash-filesystem* or **show** *flash-filesystem*: privileged EXEC command is active.

The **microcode reload** command is automatically added to your running configuration when you issue a microcode command that changes the system's default behavior of loading all processors from ROM.

In the following example, all controllers are reset, the specified microcode is loaded, and the CxBus complex is reinitialized according to the microcode configuration commands that have been written to memory:

```
Router# configure terminal
Router(config)# microcode reload
Router(config)# end
```

Displaying Microcode Image Information

To display microcode image information, use the following command in privileged EXEC mode:

Command	Purpose
`Router# show microcode`	Displays microcode information.

Using Microcode on Specific Platforms

The commands for manipulating microcode vary by platform. This section refers you to specialized configuration information found in other Cisco IOS documents.

For information on downloading microcode (Modem Firmware and Portware) into modems on Cisco access servers (like the Cisco AS5800) using the system processing engine (SPE), see the Release 12.4 *Cisco IOS Dial Technologies Configuration Guide*.

For specific information on loading CIP and CPA microcode into adapters on Cisco 7000, 7200, and 7500 series routers, see the "Configuring Cisco Mainframe Channel Connection Adapters" chapter in the "IBM Networking" part of the *Cisco IOS Bridging and IBM Networking Configuration Guide*.

Loading Microcode Images on the Cisco 12000 Internet Router

In addition to the Cisco IOS image that resides on the Internet router, each line card on the Cisco 12000 series has a Cisco IOS image. When the router is reloaded, the specified Cisco IOS image is loaded onto the GRP, and that image is automatically downloaded to all the line cards.

Normally, you want the same Cisco IOS image on the Internet router and all line cards. However, if you want to upgrade a line card with a new version of microcode for testing or to fix a defect, you may need to load a microcode system image that is different from the one on the line card. You may also need to load a new image on the line card to work around a problem that is affecting only one of the line cards.

To load a Cisco IOS image on a line card, first use the **copy tftp** privileged EXEC command to download the Cisco IOS image to a slot on one of the PCMCIA flash cards. After you have downloaded the Cisco IOS image on the flash card, use the following commands beginning in global configuration mode.

	Command	Purpose
Step 1	`Router(config)#` **`microcode`** `{`**`oc12-atm`** `|` **`oc12-pos`** `|` **`oc3-pos-4`**`}` **`flash`** *`file-id slot-number`*	Specifies the type of line card, location of the microcode image, and the slot of the line card to download the image. If the slot number is omitted, the microcode image is downloaded to all line cards.
Step 2	`Router(config)#` **`microcode reload`** *`slot-number`*	Reloads the microcode on the specified line card.
Step 3	`Router(config)#` **`exit`**	Exits configuration mode.
Step 4	`Router#` **`execute-on slot`** *`slot-number`* **`show version`** or `Router#` **`attach`** *`slot-number`*	Connects to the line card and verifies that the new Cisco IOS image is on the line card by checking the version number in the display output.

For further configuration information for Cisco 12000 series routers, see the documentation for Cisco IOS Release 11.2, Cisco IOS Release 12.0S, and Cisco IOS Release 12.2S, available on Cisco.com. For further platform specific documentation see http://www.cisco.com/univercd/cc/td/doc/product/core/.

MD5 File Validation

Feature History

Release	Modification
12.2(4)T	This feature was introduced on the 12.2 T release train.
12.0(22)S	This feature was introduced on the 12.0 S release train.

This document describes the MD5 File Validation feature in Cisco IOS Releases 12.2(4)T and 12.0(22)S. It includes the following sections:

- Feature Overview
- Supported Platforms
- Supported Standards, MIBs, and RFCs
- File Verification Tasks
- File Verification Examples
- Command Reference

Feature Overview

The MD5 File Validation feature provides a Cisco IOS software command you can use to ensure file validation using the Message Digest 5 (MD5) algorithm in the Cisco IOS File System (IFS).

The MD5 File Validation feature allows you to check the integrity of a Cisco IOS software image by comparing its MD5 checksum value against a known MD5 checksum value for the image. MD5 values are now made available on Cisco.com for all Cisco IOS software images for comparison against local system image values.

Benefits

- Provides a mechanism for users to verify that system image files are not corrupted or incomplete.
- Uses the industry-standard MD5 algorithm for improved reliability and security.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- Computes and displays the MD5 values from the Cisco IOS command-line interface (CLI); files do not have to be checked on another device.

Related Features and Technologies

- Cisco IOS File System (IFS)

Related Documents

- *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2*

Supported Platforms

For a complete list of platforms, images, and software releases that support this feature, use Cisco Feature Navigator, available through Cisco.com at:

http://www.cisco.com/go/fn

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common. The list of supported platforms is regularly updated in Cisco Feature Navigator as new platform support is added for the feature.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Supported Standards, MIBs, and RFCs

MD5 if defined in RFC 1321.

File Verification Tasks

The MD5 File Validation feature allows you to generate the MD5 checksum for the Cisco IOS image stored on your router and compare it to the posted value posted on Cisco.com to verify that the image on your router is not corrupted.

You can obtain the MD5 value for your system image from the Software Center at Cisco.com. The most convenient way to get this value is to click on the name of the file prior to download. For example, if you select the 12.2.2T4 Release for the 3640 Platform with the Enterprise Plus Feature Set, before clicking the Download button, you can click on the file name for the image (c3640-js-mz.122-2.T4.bin) and the image information will be displayed.

Image information typically includes the Release, Description, File Size, BSD Checksum, Router Checksum, Date Published, and MD5 value for the image. You should record the MD5 value for the image prior to download. However, if you do not have the MD5 value for a previously downloaded image, you can select the same image on Cisco.com (using the same process you would use to download the image) to get the MD5 value.

To perform the MD5 integrity check after transferring an image file, use the following command:

Command	Purpose
`Router#` **`verify /md5`** *`filesystem`*`:`*`filename`*	Calculates and displays the MD5 value for the software image.

Alternatively, you can specify the MD5 value in the command syntax, and the system will display a message indicating whether the values match. To specify a known MD5 value, use the following syntax:

Command	Purpose
`Router#` **`verify /md5`** *`filesystem`*`:`*`filename MD5-value`*	Checks for a match with a specified MD5 value.

A mismatch in MD5 values means that either the image is corrupt or the wrong MD5 value was entered.

File Verification Examples

In the following example, the **/md5** keyword is used to display the MD5 value for the image stored in disk1 of the device. The MD5 value shown in the last line can be compared to value provided on Cisco.com.

```
Router# verify /md5 disk1:
Verify filename []? c7200-js-mz
.................................
.................................
.................................
.................................
.................................
..............................Done!
verify /md5 (disk1:c7200-js-mz) = 0f369ed9e98756f179d4f29d6e7755d3
```

In the following example, the known MD5 value for the image is specified in the **verify** command, and the system checks the value against the stored value:

```
Router# verify /md5 disk1:c7200-js-mz ?
  WORD  Expected md5 signature
  <cr>
router# verify /md5 disk1:c7200-js-mz 0f369ed9e98756f179d4f29d6e7755d3
.................................
.................................
.................................
.................................
.................................
..............................Done!
Verified (disk1:c7200-js-mz) = 0f369ed9e98756f179d4f29d6e7755d3
```

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **verify**

Warm Upgrade

The Warm Upgrade feature provides the capability for a Cisco IOS image to read and decompress another Cisco IOS image and then transfer control to this new image. This functionality reduces the downtime of a device during planned Cisco IOS software upgrades or downgrades. The Warm Upgrade feature is complementary with the Warm Reload feature introduced in Cisco IOS Release 12.3(2)T.

Feature History for the Warm Upgrade Feature

Release	Modification
12.3(11)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Information About Warm Upgrade

To use the Warm Upgrade feature, you should understand the following concept:

- Warm Upgrade Functionality, page 2

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Warm Upgrade Functionality

The Warm Upgrade feature provides the capability for a Cisco IOS image to read and decompress another Cisco IOS image and then transfer control to this new image. This functionality reduces the downtime of a device during planned Cisco IOS software upgrades or downgrades. To perform a warm upgrade, use the **reload warm file** *url* command. The Warm Upgrade feature is complementary with the Warm Reload feature introduced in Cisco IOS Release 12.3(2)T.

Prior to the Warm Upgrade feature, a Cisco IOS image transferred control to ROM monitor mode (ROMMON) to perform a Cisco IOS software upgrade or downgrade. ROMMON, along with the help of the boot loader image, carried out the required upgrade or downgrade procedures. While this process is in progress, the networking device is down. With the introduction of the Warm Upgrade feature, packet forwarding is able to continue while the new Cisco IOS image is read and decompressed. The device is down only when the current image is overwritten with the new image, and the new image loads and reconfigures the operating system.

If a warm upgrade operation fails, the current Cisco IOS image should continue to run unless it has been partly or fully overwritten. In this case, ROMMON is allowed to load any image that is configured.

Note For cases where a Cisco IOS image is to be downgraded to an image that does not support the image verification functionality of the **reload** command, a warning message will be displayed before the warm upgrade operation is performed telling the user that the image does not have a digital signature.

How to Reload a Cisco IOS Image Using the Warm Upgrade Functionality

This section contains the following procedures:

- Reloading a Cisco IOS Image Using the Warm Upgrade Functionality, page 2 (required)
- Monitoring and Troubleshooting the Warm Upgrade Functionality, page 3 (optional)

Reloading a Cisco IOS Image Using the Warm Upgrade Functionality

Perform this task to reload a Cisco IOS image using the warm upgrade functionality.

Prerequisites

- The Warm Reload feature introduced in Cisco IOS Release 12.3(2)T must be enabled.
- The ability to upgrade or downgrade a Cisco IOS image using the Warm Upgrade feature assumes that the current Cisco IOS image supports the warm upgrade functionality. However, the new image to which the current image is being upgraded or downgraded does not need to support the warm upgrade functionality.

Restrictions

A software upgrade or downgrade using the warm upgrade functionality can only be performed if there is enough free memory in the system to accommodate a decompressed Cisco IOS image.

SUMMARY STEPS

1. **enable**
2. **reload** [**/verify** | **/noverify**] [**warm** [**file** *url*]] [**in** [*hh*:]*mm* | **at** *hh*:*mm* [*month day* | *day month*]] [**cancel**] [*text*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`reload`** `[`**`/verify`** `|` **`/noverify`**`] [`**`warm`** `[`**`file`** *`url`*`]] [`**`in`** `[`*`hh`*`:]`*`mm`* `|` **`at`** *`hh`*`:`*`mm`* `[`*`month day`* `|` *`day month`*`]] [`**`cancel`**`] [`*`text`*`]` **Example:** `Router> reload warm file flash:c3745-ipvoice-mz.12.3.11.T.bin`	Reloads the operating system. • Use the **reload warm file** *url* command to reload the operating system with a new image whose location and name is specified by the *url* argument. The reload will be performed using the warm upgrade functionality. • You must issue the **warm** keyword if you do not want to override the warm reboot functionality when you reload the router.

Monitoring and Troubleshooting the Warm Upgrade Functionality

Perform this task to monitor and troubleshoot the warm upgrade functionality.

SUMMARY STEPS

1. **show warm-reboot**
2. **debug warm-reboot**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show warm-reboot** **Example:** `Router> show warm-reboot`	Displays the statistics for attempted warm reboots.
Step 2	**debug warm-reboot** **Example:** `Router> debug warm-reboot`	Displays warm reboot debug information.

Configuration Examples for the Warm Upgrade Feature

This section provides the following configuration example:

- Reloading a Cisco IOS Image Using the Warm Upgrade Functionality: Example, page 4

Reloading a Cisco IOS Image Using the Warm Upgrade Functionality: Example

The following example shows how to reload the operating system with a new image whose location and name is tftp://9.1.0.1/c7200-p-mz.port. The reload is performed using the warm upgrade functionality.

```
Router> reload warm file tftp://9.1.0.1/c7200-p-mz.port

Proceed with reload? [confirm]
Loading c7200-p-mz.port from 9.1.0.1 (via Ethernet5/0):!!!
[OK - 15323964 bytes]

Decompressing the image :### [OK]

02:37:42:%SYS-5-RELOAD:Reload requested by console. Reload Reason:Reload Command.
             Restricted Rights Legend
.
.
.
Press RETURN to get started!

00:00:12:%LINK-3-UPDOWN:Interface Ethernet5/0, changed state to up
00:00:12:%LINK-3-UPDOWN:Interface Ethernet5/1, changed state to up
00:00:12:%LINK-3-UPDOWN:Interface Ethernet5/2, changed state to up
00:00:12:%LINK-3-UPDOWN:Interface Ethernet5/3, changed state to up
00:00:12:%LINK-3-UPDOWN:Interface FastEthernet6/0, changed state to up
00:00:12:%LINK-3-UPDOWN:Interface FastEthernet6/1, changed state to up
00:00:12:%SYS-5-CONFIG_I:Configured from memory by console
00:00:13:%SYS-5-RESTART:System restarted --
00:00:13:%SYS-6-BOOTTIME:Time taken to reboot after reload =   25 seconds
00:00:14:%LINEPROTO-5-UPDOWN:Line protocol on Interface Ethernet5/0, changed state to up
00:00:14:%LINEPROTO-5-UPDOWN:Line protocol on Interface Ethernet5/1, changed state to down
00:00:14:%LINEPROTO-5-UPDOWN:Line protocol on Interface Ethernet5/2, changed state to down
00:00:14:%LINEPROTO-5-UPDOWN:Line protocol on Interface Ethernet5/3, changed state to down
00:00:14:%LINEPROTO-5-UPDOWN:Line protocol on Interface FastEthernet6/0, changed state to
down
00:00:14:%LINEPROTO-5-UPDOWN:Line protocol on Interface FastEthernet6/1, changed state to
down
```

```
00:00:14:%LINEPROTO-5-UPDOWN:Line protocol on Interface Fddi4/0, changed state to down
00:00:14:%LINK-5-CHANGED:Interface Fddi4/0, changed state to administratively down
00:00:14:%LINK-5-CHANGED:Interface Ethernet5/1, changed state to administratively down
00:00:14:%LINK-5-CHANGED:Interface Ethernet5/2, changed state to administratively down
00:00:14:%LINK-5-CHANGED:Interface Ethernet5/3, changed state to administratively down
00:00:14:%LINK-5-CHANGED:Interface FastEthernet6/0, changed state to administratively down
00:00:14:%LINK-5-CHANGED:Interface FastEthernet6/1, changed state to administratively down
```

Additional References

The following sections provide references related to the Warm Upgrade feature.

Related Documents

Related Topic	Document Title
Additional information on rebooting your router	*The chapter "Rebooting" in the section "File Management" in the Cisco IOS Configuration Fundamentals and Network Management Configuration Guide, Release 12.3*
Additional booting commands	*Cisco IOS Configuration Fundamentals and Network Management Command Reference*, Release 12.3T

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals*

Command Reference at
http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at
http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **debug warm-reboot**
- **reload**

Maintaining System Memory

Maintaining System Memory

This chapter describes how to maintain and use the different types of memory on your router. This document applies to Cisco IOS Release 12.2.

For a complete description of the memory commands mentioned in this chapter, refer to the "Router Memory Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Understanding Memory Types and Functions

Your router has many different locations where it can store images, configuration files, and microcode. Refer to your hardware documentation for details on which types of memory your routing device contains, where files can be stored (saved), and where images and boot images are located by default. This section provides information on the following memory types:

- DRAM
- EPROM
- NVRAM
- Flash Memory

DRAM

Dynamic random-access memory (DRAM) contains two types of memory:

- Primary, main, or processor memory, which is reserved for the CPU to execute Cisco IOS software and to hold the running configuration and routing tables.
- Shared, packet, or I/O memory, which buffers data transmitted or received by the router's network interfaces.

On the Cisco 3600 series routers, you can use the **memory-size iomem** command to configure the proportion of DRAM devoted to main memory and to shared memory.

DRAM often comes on dual in-line memory modules (DIMMs).

EPROM

Erasable programmable read-only memory (EPROM) is often referred to simply as ROM. On Cisco devices, the EPROM often contains the following:

- ROM Monitor software, which provides a user interface for troubleshooting the ROM.
- The boot loader/helper software, which helps the router boot when it cannot find a valid Cisco IOS image in Flash memory.

NVRAM

Non-volatile random-access-memory (NVRAM) stores the following information:

- Startup configuration file for every platform except Class A Flash file system platforms (for Class A Flash file system platforms, the location of the startup configuration depends on the CONFIG_FILE Environment Variable).
- The software configuration register, which is used to determine which image to use when booting the router.

Flash Memory

Flash memory stores the Cisco IOS software image. On most platforms, it can store boot-images and/or configuration files.

Depending on the hardware platform, Flash memory might be available as EPROM, single in-line memory modules (SIMMs), dual in-line memory modules (DIMMs), or Flash memory cards. Check the appropriate hardware installation and maintenance guide for information about types of Flash memory available on a specific platform.

Depending on the platform, Flash memory is available in the following forms:

- Internal Flash memory
  - Internal Flash memory often contains the system image.
  - Some platforms have two or more banks of Flash memory on one in-line memory module (in other words, on one SIMM). If the SIMM has two banks, it is sometimes referred to as *dual-bank Flash memory*. The banks can be partitioned into separate logical devices. See the "Partitioning Flash Memory" section for information about how to partition Flash memory.
- Bootflash
  - Bootflash often contains the boot image.
  - Bootflash sometimes contains the ROM Monitor.
- Flash memory PC cards or PCMCIA cards

  A Flash memory card that is inserted in to a Personal Computer Memory Card International Association (PCMCIA) slot. This card is used to store system images, boot images, and configuration files.

Note Because some platforms, such as the Cisco 3600 series and Cisco the 7000 family, can boot images and load configuration files from several locations, these systems use special ROM monitor environment variables to specify the location and filename of images and configuration files that the router is to use for various functions.

Many Cisco routers load the system image from flash storage into RAM in order to run the Cisco IOS. However, some platforms, such as the Cisco 1600 Series and Cisco 2500 Series, execute the Cisco IOS operation system directly from Flash memory. These platforms are run-from-Flash memory systems.

If you want to partition Flash memory, you must use a relocatable image. Relocatable images can be run from any location in Flash and can download images to any location. If you are upgrading from a nonrelocatable image to a relocatable image, you must erase Flash memory during the download so that the image is downloaded as the first file in Flash memory. All images for run-from-Flash platforms from Cisco IOS Release 11.0 and later are relocatable. See the "Image Naming Conventions" section in the "Loading and Maintaining System Images" chapter to determine if your images are run-from-Flash images or are relocatable.

Flash memory provides write protection against accidental erasing or reprogramming. Some platforms have a write-protect jumper which can be removed to prevent reprogramming of Flash memory. You must install the jumper when programming is required. Some platforms have write protect switched on Flash memory cards that you can use to protect data. You must set the switch to *unprotected* to write data to the Flash memory card. Refer to your hardware documentation for information on security jumpers and write protect switches.

Note The internal Flash and Flash memory cards of a system cannot be used as a contiguous bank of Flash memory.

Maintaining System Memory Task List

You can perform the tasks related to Flash memory in the following sections:

- Displaying System Memory Information
- Reallocating DRAM Memory for the Cisco 3600 Series
- Partitioning Flash Memory
- Using Flash Load Helper to Upgrade Software on Run-from-Flash Systems
- Formatting Flash Memory

The tasks in this chapter assume that you have a minimal configuration that you want to modify.

Displaying System Memory Information

Use the following commands in EXEC mode to display information about system memory:

Command	Purpose
Router# **show** *flash-filesystem*: [**all** | **chips** | **filesys**]	Lists information about Flash memory for Class A file systems.
Router# **show** *flash-filesystem*: [**partition** *number*] [**all** | **chips** | **detailed** | **err** | **summary**]	Lists information about Flash memory for Class B file systems.
Router# **show** *flash-filesystem*:	Lists information about Flash memory for Class C file systems.
Router# **show file systems**	Lists the names of the file systems currently supported on the router.

Partitioning Flash Memory

On most Class B Flash file systems, you can partition banks of Flash memory into separate, logical devices so that the router can hold and maintain two or more different software images. This partitioning allows you to write software into Flash memory while running software in another bank of Flash memory.

Systems that Support Partitioning

To partition Flash memory, you must have at least two banks of Flash memory; a bank is a set of 4 chips. This requirement includes systems that support a single SIMM that has two banks of Flash memory. The minimum partition size is the size of a bank.

Note The CiscoFlash MIB variables support partitioned Flash.

Benefits of Partitioning Flash Memory

Partitioning Flash memory provides the following benefits:

- For any system, partitioning—rather than having one logical Flash memory device—provides a cleaner way of managing different files in Flash memory, especially if the Flash memory size is large.
- For systems that execute code out of Flash memory, partitioning allows you to download a new image into the file system in one Flash memory bank while an image is being executed from the file system in the other bank. The download is simple and causes no network disruption or downtime. After the download is complete, you can switch over to the new image at a convenient time.
- One system can hold two different images, one image acting as a backup for the other. Therefore, if a downloaded image fails to boot for some reason, the earlier running, good image is still available. Each bank is treated as a separate device.

Flash Load Helper Versus Dual Flash Bank

Flash load helper is a software option that enables you to upgrade system software on run-from-Flash systems that have a single bank of Flash memory. It is a lower-cost software upgrade solution than dual-bank Flash, which requires two banks of Flash memory on one SIMM. Flash load helper is only available on run-from-Flash platforms, such as the Cisco 2500 series, Cisco 3000, and Cisco 5200.

You might use Flash load helper rather than partitioning Flash into two banks for one of the following reasons:

- If you want to download a new file into the same bank from which the current system image is executing.
- If you want to download a file that is larger than the size of a bank, and hence want to switch to a single-bank mode.
- If you have only one single-bank Flash SIMM installed. In this case, Flash load helper is the best option for upgrading your software.

See the “Using Flash Load Helper to Upgrade Software on Run-from-Flash Systems” section for information about using Flash load helper.

Partitioning Flash Memory

To partition Flash memory, use one of the forms of the following command in global configuration mode:

Command	Purpose
Router(config)# **partition flash** *partitions* [*size1 size2*]	Partitions Flash memory.
Router(config)# **partition** *flash-filesystem***:** [*number-of-partitions*] [*partition-size*]	Partitions Flash memory on the Cisco 1600 and 3600 series.

This task will succeed only if the system has at least two banks of Flash and the partitioning does not cause an existing file in Flash memory to be split across the partitions.

For all platforms except the Cisco 1600 series and Cisco 3600 series, Flash memory can only be partitioned into two partitions.

For the Cisco 1600 series and Cisco 3600 series, the number of partitions that you can create in a Flash memory device equals the number of banks in the device. Enter the **show** *flash-filesystem***: all** command to view the number of banks on the Flash memory device. The number of partition size entries you set must be equal to the number of specified partitions. For example, the **partition slot0: 2 8 8** command configures two partitions to be 8 MB in size each. The first 8 corresponds to the first partition; the second 8 corresponds to the second partition.

Note To remove the partition, use the **no partition** command.

Using Flash Load Helper to Upgrade Software on Run-from-Flash Systems

Flash load helper is a software option that enables you to upgrade system software on run-from-Flash systems that have a single bank of Flash memory. It is a lower-cost software upgrade solution than dual-bank Flash, which requires two banks of Flash memory on one SIMM.

The Flash load helper software upgrade process is simple and does not require additional hardware; however, it does require some brief network downtime. A system image running from Flash can use Flash load helper only if the boot ROMs support Flash load helper. Otherwise, you must perform the Flash upgrade manually. See the “Manually Boot from Flash Memory” section.

Flash load helper is an automated procedure that reloads the ROM-based image, downloads the software to Flash memory, and reboots to the system image in Flash memory. Flash load helper performs checks and validations to maximize the success of a Flash upgrade and minimize the chance of leaving Flash memory either in an erased state or with a file that cannot boot.

In run-from-Flash systems, the software image is stored in and executed from the Flash EPROM rather than from RAM. This method reduces memory cost. A run-from-Flash system requires enough Flash EPROM to hold the image and enough main system RAM to hold the routing tables and data structures. The system does not need the same amount of main system RAM as a run-from-RAM system because the full image does not reside in RAM. Run-from-Flash systems include the Cisco 2500 series and some Cisco 3000 series.

Flash Load Helper Features

Flash load helper performs the following functions:

- Confirms access to the specified source file on the specified server before erasing Flash memory and reloading to the ROM image for the actual upgrade.
- Warns you if the image being downloaded is not appropriate for the system.
- Prevents reloads to the ROM image for a Flash upgrade if the system is not set up for automatic booting and the user is not on the console terminal. In the event of a catastrophic failure during the upgrade, Flash load helper can bring up the boot ROM image as a last resort rather than forcing the system to wait at the ROM monitor prompt for input from the console terminal.
- Retries Flash downloads automatically up to six times. The retry sequence is as follows:
  - First try
  - Immediate retry
  - Retry after 30 seconds
  - Reload ROM image and retry
  - Immediate retry
  - Retry after 30 seconds
- Allows you to save any configuration changes made before you exit out of the system image.
- Notifies users logged in to the system of the impending switch to the boot ROM image so that they do not lose their connections unexpectedly.
- Logs console output during the Flash load helper operation into a buffer that is preserved through system reloads. You can retrieve the buffer contents from a running image. The output is useful when console access is unavailable or a failure occurs in the download operation.

Flash load helper can also be used on systems with multiple banks of Flash memory that support Flash memory partitioning. Flash load helper enables you to download a new file into the same partition from which the system is executing an image.

For information about how to partition multiple banks of Flash memory so your system can hold two different images, see the "Partitioning Flash Memory" section.

Downloading Files Using the Flash Load Helper

To download a new file to Flash memory using Flash load helper, check to make sure that your boot ROMs support Flash load helper and then use one of the following commands in privileged EXEC mode:

Command	Purpose
Router# **copy tftp: flash:** Router# **copy rcp: flash:** Router# **copy ftp: flash:**	Loads the specified file to Flash memory.

The following error message displays if you are in a Telnet session and the system is set for manual booting (the boot bits in the configuration register are zero):

```
ERR: Config register boot bits set for manual booting
```

In case of any catastrophic failure in the Flash memory upgrade, this error message helps to minimize the chance of the system going down to ROM monitor mode and being taken out of the remote Telnet user's control.

The system tries to bring up at least the boot ROM image if it cannot boot an image from Flash memory. Before reinitiating the **copy:** command, you must set the configuration register boot field to a nonzero value, using the **config-register** global configuration command.

The **copy** command initiates a series of prompts to which you must provide responses. The dialog is similar to the following:

```
Router# copy tftp: flash:

*************************** NOTICE *******************************
Flash load helper v1.0
This process will accept the TFTP copy options and then terminate
the current system image to use the ROM based image for the copy.
Router functionality will not be available during that time. If
you are logged in via telnet, this connection will terminate. Users
with console access can see the results of the copy operation.
******************************************************************

There are active users logged into the system.

Proceed? [confirm] y
System flash directory:
File Length  Name/status
1    2251320 abc/igs-kf.914
[2251384 bytes used, 1942920 available, 4194304 total]
Address or name of remote host [255.255.255.255]? 172.16.1.111
Source file name? abc/igs-kf.914
Destination file name [default = source name]? <Return>
Accessing file 'abc/igs-kf.914' on 172.16.1.111....
Loading from 172.16.13.111:
Erase flash device before writing? [confirm] n
File 'abc/igs-kf.914' already exists; it will be invalidated!
```

```
Invalidate existing copy of 'abc/igs-kf.914' in flash memory? [confirm] y
Copy 'abc/igs-kf.914' from TFTP server
as 'abc/igs-kf.914' into Flash WITHOUT erase? y

%SYS-5-RELOAD: Reload requested
%FLH: rxboot/igs-kf.914r from 172.16.1.111 to flash...
```

The Flash Load Helper operation verifies the request from the running image by trying to copy a single block from the remote server. Then the Flash load helper is executed, causing the system to reload to the ROM-based system image. If the file does not seem to be a valid image for the system, a warning is displayed and a separate confirmation is sought from you.

If the configuration has been modified but not yet saved, you are prompted to save the configuration:

```
System configuration has been modified. Save? [confirm]
```

Users with open Telnet connections are notified of the system reload, as follows:

```
**System going down for Flash upgrade**
```

If the copy process fails, the copy operation is retried up to three times. If the failure happens in the middle of a copy operation so that only part of the file has been written to Flash memory, the retry does not erase Flash memory unless you specified an erase operation. The partly written file is marked as deleted, and a new file is opened with the same name. If Flash memory runs out of free space in this process, the copy operation is terminated.

After Flash load helper finishes copying (whether the copy operation is successful or not), it automatically attempts an automatic or a manual boot, depending on the value of bit zero of the configuration register boot field according to the following:

- If bit zero equals 0, the system attempts a default boot from Flash memory to load up the first bootable file in Flash memory. This default boot is equivalent to a manual **boot flash** command at the ROM monitor prompt.
- If bit zero equals 1, the system attempts to boot based on the boot configuration commands. If no boot configuration commands exist, the system attempts a default boot from Flash memory; that is, it attempts to load the first bootable file in Flash memory.

To view the system console output generated during the Flash load helper operation, use the image that has been booted up after the Flash memory upgrade. Use the following command in privileged EXEC mode:

Command	Purpose
`Router#` **`more flh:logfile`**	View the console output generated during the Flash load helper operation.

If you are a remote Telnet user performing the Flash upgrade without a console connection, this task allows you to retrieve console output when your Telnet connection has terminated due to the switch to the ROM image. The output indicates what happened during the download, and is particularly useful if the download fails.

Formatting Flash Memory

On Class A and Class C Flash file systems, you can format Flash memory. Formatting erases all information in Flash memory.

On the Cisco 7000 family, you must format a new Flash memory card before using it in a PCMCIA slot.

Flash memory cards have sectors that can fail. You can reserve certain Flash memory sectors as "spares" for use when other sectors fail. Use the **format** command to specify between 0 and 16 sectors as spares. If you reserve a small number of spare sectors for emergencies, you do not waste space because you can use most of the Flash memory card. If you specify zero spare sectors and some sectors fail, you must reformat the Flash memory card and thereby erase all existing data.

The format operation requires at least Cisco IOS Release 11.0 system software.

Flash Memory Formatting Process

Caution The following formatting procedure erases all information in Flash memory. To prevent the loss of important data, proceed carefully.

Use the following procedure to format Flash memory. If you are formatting internal Flash memory, such as bootflash, you can skip the first step. If you are formatting a Flash memory card, complete both steps.

Step 1 Insert the new Flash memory card into a PCMCIA slot. Refer to instructions on maintaining the router and replacing PCMCIA cards in your router's hardware documentation for instructions on performing this step.

Step 2 Format Flash memory.

To format Flash memory, use the following EXEC mode command:

Command	Purpose
Router# **format** [**spare** *spare-number*] *device1*: [[*device2*:][*monlib-filename*]]	Formats Flash memory.

The following example shows the **format** command that formats a Flash memory card inserted in slot 0.

```
Router# format slot0:
Running config file on this device, proceed? [confirm]y
All sectors will be erased, proceed? [confirm]y
Enter volume id (up to 31 characters): <Return>
Formatting sector 1 (erasing)
Format device slot0 completed
```

When the router returns you to the EXEC prompt, the new Flash memory card is successfully formatted and ready for use.

Recovering from Locked Blocks

To recover from locked blocks, reformat the Flash memory card. A locked block of Flash memory occurs when power is lost or a Flash memory card is unplugged during a write or erase operation. When a block of Flash memory is locked, it cannot be written to or erased, and the operation will consistently fail at a particular block location. The only way to recover from locked blocks is by reformatting the Flash memory card with the **format** command.

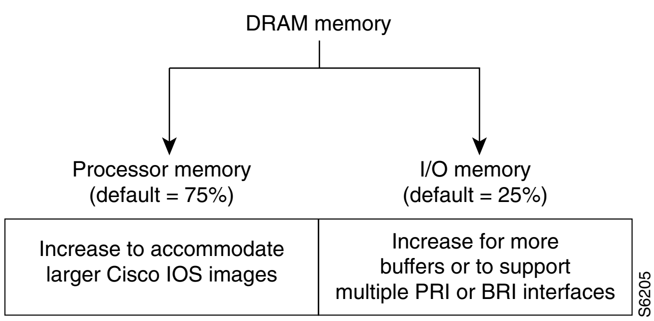
Caution Formatting a Flash memory card to recover from locked blocks will cause existing data to be lost.

Reallocating DRAM Memory for the Cisco 3600 Series

DRAM memory in Cisco 3600 series routers is organized as one contiguous address space divided between processor memory and I/O memory. Depending on the type and number of network interfaces you have configured in the router, you may need to reallocate the DRAM memory partitioned to processor memory and I/O memory.

Cisco manufacturing configures most Cisco 3600 series routers to have 25 percent of the address space allocated to I/O memory and 75 percent allocated to processor memory. But for customer orders that require two or more ISDN PRI interfaces, DRAM memory is configured to provide 40 percent of the address space for I/O memory and 60 percent for processor memory. (See Figure 11.) Cisco Systems performs these DRAM memory adjustments before it ships each router.

Figure 11 Components and Uses of DRAM Memory for Cisco 3600 Series Routers



Note Routers running two or more ISDN PRI interfaces or 12 or more ISDN BRI interfaces require a DRAM memory configuration of 40 percent I/O memory and 60 percent processor memory.

However, there are cases where you may have to manually reallocate the DRAM memory split between processor memory and I/O memory after you have received a router from Cisco Systems.

For example, suppose you receive a Cisco 3640 router with the following running configuration:

- 2 Ethernet and 2 WAN interface card
- 8-port ISDN BRI with an NT1 network module
- IP feature set
- 16 MB of DRAM memory (by default, processor memory = 75%, I/O memory = 25%)
- 4 MB of Flash memory

Later, however, you add a 4-port ISDN BRI network module to the router. You now have 12 ISDN BRI interfaces running on the router. At this point, you must use the **memory-size iomem** command to configure 40 percent of the address space for I/O memory and 60 percent for processor memory.

To view your current mix of processor and I/O memory and reassign memory distribution accordingly, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# show version`	Displays the total amount of memory loaded on the router.
Step 2	`Router# show memory`[1]	Displays the amount of free memory.
Step 3	`Router# configure terminal`	Enters global configuration mode.
Step 4	`Router(config)# memory-size iomem` *`I/O-memory-percentage`*[2]	Allocates processor memory and I/O memory.
Step 5	`Router(config)# exit`	Exits global configuration mode.
Step 6	`Router# copy system:running-config nvram:startup-config`	Saves the configuration to NVRAM.
Step 7	`Router# reload`	Reloads the router to run the new image.

1. The Free(b) column in the **show memory** command's output shows how much I/O memory is available.
2. The default is 40 percent for I/O memory and 60 percent for processor memory.

Valid I/O memory percentage values are 10, 15, 20, 25, 30, 40 (the default), and 50. I/O memory size is the specified percentage of total memory size, rounded down to the nearest multiple of 1 MB. A minimum of 4 MB of memory is required for I/O memory. The remaining memory is processor memory.

The **memory-size iomem** command does not take effect until you save it to NVRAM using the **copy system:running-config nvram:startup-config** EXEC command and reload the router. However, when you enter the command, the software checks whether the new memory distribution leaves enough processor memory for the currently running Cisco IOS image. If not, the following message appears:

```
Warning: Attempting a memory partition that does not provide enough Processor memory for
the current image.If you write memory now, this version of software may not be able to
run.
```

When you enter the **reload** command to run a new image, the software calculates the new processor and I/O memory split. If there is not enough processor memory, it automatically reduces I/O memory to an alternative setting to load the image. If there is still not enough processor memory for the image to run, then you do not have enough DRAM.

Reallocate Processor Memory and I/O Memory Example

The following example allocates 40 percent of DRAM to I/O memory and the remaining 60 percent to processor memory. The example views the current allocation of memory, changes the allocation, saves the allocation, and reloads the router so the changes can take effect. In the **show memory** command output, the Free(b) column shows how much I/O memory is available:

```
Router# show memory
                Head   Total(b)     Used(b)     Free(b)   Lowest(b) Largest(b)
Processor   60913730    3066064      970420     2095644     2090736    2090892
      I/O     C00000    4194304     1382712     2811592     2811592    2805492
--More--

Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# memory-size iomem 40
Router(config)# exit
Router#
Router# copy system:running-config nvram:startup-config
Building configuration...
```

```
[OK]

Router# reload

rommon > boot
program load complete, entry point: 0x80008000, size: 0x32ea24
Self decompressing the image :
###########################################################################################
###########################################################################################
################################################################### [OK]
```

Using Memory Scan on the Cisco 7500 Series

On Cisco 7500 series routers (including 7000 series with the RSP7000 card upgrade), a memory scanning feature is available. This feature adds a low-priority background process that searches all installed dynamic random-access memory (DRAM) for possible parity errors. If errors are found in memory areas that are not in use, this feature attempts to scrub (remove) the errors. The time to complete one memory scan and scrub cycle can range from 10 minutes to several hours, depending on the amount of installed memory. The impact of the Memory Scan feature on the central processing unit (CPU) is minimal. The feature can be controlled and monitored with the new **memory scan** and **show memory scan** command-line interface (CLI) commands.

The Memory Scan feature does not discriminate against different information types in DRAM; that is, it perceives text, data, and heap information in the same way. The feature continues to work when a memory cell is busy, although it might respond differently to errors found in different areas. The feature responds to errors in one or more of the following ways:

- A message is logged for all errors found. Each message contains an explanation of the error and suggests corrective action if applicable.
- For errors in heap storage control blocks, attempts are made to scrub errors in the free blocks. If an error is scrubbed, no further action occurs, but there is an entry in the error log. If it is not scrubbed, the block that contains the error is linked to a bad-memory list which will not be allocated to users. If the memory block is large, the block is split and only a small portion containing the error is linked to a bad-memory list.
- For errors in a busy block, or in other areas such as text or data, an error message is produced but no further action is taken, preventing damage to living data.

Configuring and Verifying Memory Scan

Use the **memory scan** command in global configuration mode to enable the feature.

Use the **more system:running-configuration** command in privileged EXEC mode to verify that memory scan appears in the running configuration.

Use the **show memory scan** command to monitor the number and type of parity errors on your system. Use the **show memory scan** command in privileged EXEC mode. In the following example, the feature is enabled and no parity errors are found:

```
Router# show memory scan
Memory scan is on.
No parity error has been detected.
```

If the Memory Scan feature has not been configured, or has been turned off, the **show memory scan** command generates a report. In the following example, Memory Scan is turned off:

```
Router# show memory scan
Memory scan is off
No parity error has been detected.
```

If errors are detected in the system, the **show memory scan** command generates an error report. In the following example, Memory Scan detected a parity error:

```
Router# show memory scan
Memory scan is on.
Total Parity Errors 1.
Address BlockPtr BlckSize Disposit   Region Timestamp
6115ABCD 60D5D090   9517A4  Scrubed    Local 16:57:09 UTC Thu Mar 18
```

For an explanation of the error report fields, see the full details on the **show memory scan** command in the "Router Memory Commands" chapter of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Memory Leak Detector

The Memory Leak Detector feature is a tool that can be used to detect memory leaks on a router that is running Cisco IOS software. The Memory Leak Detector feature is capable of finding leaks in all memory pools, packet buffers, and chunks.

Feature History for Memory Leak Detector

Release	Modification
12.3(8)T1	This feature was introduced.
12.2(25)S	This feature was integrated into Cisco IOS Release 12.2(25)S.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Information About Memory Leak Detector

Before using the Memory Leak Detector feature, you should understand the following concepts:

- Memory Leaks, page 2
- Memory Leak Detection, page 2

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Memory Leaks

Memory leaks are static or dynamic allocations of memory that do not serve any useful purpose. Although technology is available for detection of leaks among statically allocated memory, in this document the focus is on memory allocations that are made dynamically.

Memory Leak Detection

From the detection point of view, leaks among the dynamically allocated memory blocks can be classified into the following three types:

- Type 1 leaks have no references. These blocks of memory can not be accessed.
- Type 2 leaks are part of one or more cycles of allocations but none of the blocks in these cycles is accessible from outside of the cycles. Blocks within each cycle have references to other elements in the cycle(s). An example of a Type 2 leak is a circular list that is not needed anymore. Though individual elements are reachable, the circular list is not reachable.
- Type 3 leaks are accessible or reachable but are not needed, for example, elements in data structures that are not needed anymore. A subclass of Type 3 leaks are those where allocations are made but never written to. You can look for these subclass leaks using the **show memory debug reference unused** command.

The Memory Leak Detector feature provides the technology to detect Type 1 and Type 2 memory leaks.

The Memory Leak Detector feature works in the following two modes:

- Normal mode—Where memory leak detector uses memory to speed up its operations.
- Low memory mode—Where memory leak detector runs without attempting to allocate memory.

Low memory mode is considerably slower than the normal mode and can handle only blocks. There is no support for chunks in low memory mode. Low memory mode is useful when there is little or no memory available on the router.

The memory leak detector has a simple interface and can be invoked by the command line interface (CLI) at any time to get a report of memory leaks. For testing purposes, you can perform all tests, then invoke memory leak detector to get a report on leaks. If you are interested only in leaks generated by your test cases alone, memory leak detector has an incremental option, which can be enabled at the start of testing. After testing completes, you can get a report on only the leaks that occurred after the incremental option was enabled.

To reduce false alarms, it is mandatory that memory leak detector be invoked multiple times and that only leaks that consistently appear in all reports be interpreted as leaks. This is especially true for packet buffer leaks.

Note When submitting defects based on the reports of memory leak detector, please add “memleak-detection” to the attribute field of the defect report.

How to Use Memory Leak Detector

This section contains the following procedures:

- Displaying Memory Leak Information, page 3
- Setting the Memory Debug Incremental Starting Time, page 8
- Displaying Memory Leak Information Incrementally, page 8

Displaying Memory Leak Information

This task describes how to display detected memory leak information.

SUMMARY STEPS

1. **enable**
2. **show memory debug leaks** [**chunks** | **largest** | **lowmem** | **summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show memory debug leaks** or **show memory debug leaks** [**chunks**] or **show memory debug leaks** [**largest**] or **show memory debug leaks** [**lowmem**] or **show memory debug leaks** [**summary**] **Example:** `Router# show memory debug leaks` or **Example:** `Router# show memory debug leaks chunks` or **Example:** `Router# show memory debug leaks largest` or **Example:** `Router# show memory debug leaks lowmem` or **Example:** `Router# show memory debug leaks summary`	Invokes normal mode memory leak detection and displays detected memory leaks. It does not detect memory leaks in chunks. or (Optional) Invokes normal mode memory leak detection and displays detected memory leaks in chunks. or (Optional) Invokes memory leak detection and displays the top ten leaking allocator_pcs and total amount of memory that they have leaked. Additionally, each time this command is invoked it remembers the previous invocation's report and compares it to the current invocation's report. or (Optional) Invokes low memory mode memory leak detection and displays detected memory leaks. The amount of time taken for analysis is considerably greater than that of normal mode. The output for this command is similar to the **show memory debug leaks** command. or (Optional) Invokes normal mode memory leak detection and displays detected memory leaks based on allocator_pc and then on the size of the block.

Examples

This section provides the following output examples:

- Sample Output for the show memory debug leaks Command, page 5
- Sample Output for the show memory debug leaks chunks Command, page 5
- Sample Output for the show memory debug leaks largest Command, page 6
- Sample Output for the show memory debug leaks summary Command, page 7

Sample Output for the show memory debug leaks Command

The following example shows output from the **show memory debug leaks** command with no optional keywords specified:

```
Router# show memory debug leaks

Adding blocks for GD...

                  PCI memory
Address     Size   Alloc_pc  PID  Name

                  I/O memory
Address     Size   Alloc_pc  PID  Name

                  Processor memory
Address     Size   Alloc_pc  PID  Name
62DABD28      80 60616750  -2   Init
62DABD78      80 606167A0  -2   Init
62DCF240      88 605B7E70  -2   Init
62DCF298      96 605B7E98  -2   Init
62DCF2F8      88 605B7EB4  -2   Init
62DCF350      96 605B7EDC  -2   Init
63336C28     104 60C67D74  -2   Init
63370D58      96 60C656AC  -2   Init
633710A0     304 60C656AC  -2   Init
63B2BF68      96 60C659D4  -2   Init
63BA3FE0   32832 608D2848  104  Audit Process
63BB4020   32832 608D2FD8  104  Audit Process
```

Table 1 describes the significant fields shown in the display.

Table 1 show memory debug leaks Field Descriptions

Field	Description
Address	Hexadecimal address of the leaked block.
Size	Size of the leaked block (in bytes).
Alloc_pc	Address of the system call that allocated the block.
PID	The process identifier of the process that allocated the block.
Name	The name of the process that allocated the block.

Sample Output for the show memory debug leaks chunks Command

The following example shows output from the **show memory debug leaks chunks** command:

```
Router# show memory debug leaks chunks

Adding blocks for GD...

                  PCI memory
Address     Size   Alloc_pc  PID  Name

Chunk Elements:
Address   Size  Parent    Name

                  I/O memory
Address     Size   Alloc_pc  PID  Name

Chunk Elements:
```

```
Address   Size   Parent   Name

                 Processor memory
Address    Size   Alloc_pc  PID  Name
62DABD28      80 60616750  -2   Init
62DABD78      80 606167A0  -2   Init
62DCF240      88 605B7E70  -2   Init
62DCF298      96 605B7E98  -2   Init
62DCF2F8      88 605B7EB4  -2   Init
62DCF350      96 605B7EDC  -2   Init
63336C28     104 60C67D74  -2   Init
63370D58      96 60C656AC  -2   Init
633710A0     304 60C656AC  -2   Init
63B2BF68      96 60C659D4  -2   Init
63BA3FE0   32832 608D2848  104  Audit Process
63BB4020   32832 608D2FD8  104  Audit Process

Chunk Elements:
Address   Size   Parent   Name
62D80DA8    16 62D7BFD0 (Managed Chunk )
62D80DB8    16 62D7BFD0 (Managed Chunk )
62D80DC8    16 62D7BFD0 (Managed Chunk )
62D80DD8    16 62D7BFD0 (Managed Chunk )
62D80DE8    16 62D7BFD0 (Managed Chunk )
62E8FD60   216 62E8F888 (IPC Message He)
```

Table 2 describes the significant fields shown in the display.

Table 2 show memory debug leaks chunks Field Descriptions

Field	Description
Address	Hexadecimal address of the leaked block.
Size	Size of the leaked block (in bytes).
Alloc_pc	Address of the system call that allocated the block.
PID	The process identifier of the process that allocated the block.
Name	The name of the process that allocated the block.
Size	(Chunk Elements) Size of the leaked element (bytes).
Parent	(Chunk Elements) Parent chunk of the leaked chunk.
Name	(Chunk Elements) The name of the leaked chunk.

Sample Output for the show memory debug leaks largest Command

The following example shows output from the **show memory debug leaks largest** command:

```
Router# show memory debug leaks largest

Adding blocks for GD...

                 PCI memory
Alloc_pc    total leak size

                 I/O memory
Alloc_pc    total leak size

                 Processor memory
Alloc_pc    total leak size
608D2848    32776     inconclusive
```

```
608D2FD8    32776     inconclusive
60C656AC    288       inconclusive
60C67D74    48        inconclusive
605B7E98    40        inconclusive
605B7EDC    40        inconclusive
60C659D4    40        inconclusive
605B7E70    32        inconclusive
605B7EB4    32        inconclusive
60616750    24        inconclusive
```

The following example shows output from the second invocation of the **show memory debug leaks largest** command:

```
Router# show memory debug leaks largest

Adding blocks for GD...

                 PCI memory
Alloc_pc    total leak size

                 I/O memory
Alloc_pc    total leak size

                 Processor memory
Alloc_pc    total leak size
608D2848    32776
608D2FD8    32776
60C656AC    288
60C67D74    48
605B7E98    40
605B7EDC    40
60C659D4    40
605B7E70    32
605B7EB4    32
60616750    24
```

Sample Output for the show memory debug leaks summary Command

The following example shows output from the **show memory debug leaks summary** command:

```
Router# show memory debug leaks summary

Adding blocks for GD...

                 PCI memory

Alloc PC        Size     Blocks      Bytes    What

                 I/O memory

Alloc PC        Size     Blocks      Bytes    What

                 Processor memory

Alloc PC        Size     Blocks      Bytes    What

0x605B7E70 0000000032 0000000001 0000000032    Init
0x605B7E98 0000000040 0000000001 0000000040    Init
0x605B7EB4 0000000032 0000000001 0000000032    Init
0x605B7EDC 0000000040 0000000001 0000000040    Init
0x60616750 0000000024 0000000001 0000000024    Init
0x606167A0 0000000024 0000000001 0000000024    Init
```

```
0x608D2848 0000032776 0000000001 0000032776    Audit Process
0x608D2FD8 0000032776 0000000001 0000032776    Audit Process
0x60C656AC 0000000040 0000000001 0000000040    Init
0x60C656AC 0000000248 0000000001 0000000248    Init
0x60C659D4 0000000040 0000000001 0000000040    Init
0x60C67D74 0000000048 0000000001 0000000048    Init
```

Table 3 describes the significant fields shown in the display.

Table 3 show memory debug leaks summary Field Descriptions

Field	Description
Alloc PC	Address of the system call that allocated the block.
Size	Size of the leaked block.
Blocks	Number of blocks leaked.
Bytes	Total amount of memory leaked.
What	Name of the process that owns the block.

Setting the Memory Debug Incremental Starting Time

This task describes how to set the starting time for incremental analysis of memory leaks. For incremental analysis, you can define a starting point by using the **set memory debug incremental starting-time** command. When the starting time is set, only memory allocated after the starting time will be considered for reporting as leaks.

SUMMARY STEPS

1. **enable**
2. **set memory debug incremental starting-time**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**set memory debug incremental starting-time** **Example:** `Router# set memory debug incremental starting-time`	Sets the starting time for incremental analysis to the time when the command is issued.

Displaying Memory Leak Information Incrementally

This task describes how to display memory leak information after a starting time has been established.

SUMMARY STEPS

1. **enable**
2. **set memory debug incremental starting-time**
3. **show memory debug incremental** {**allocations** | **leaks** [**lowmem**] | **status**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`set memory debug incremental starting-time`** **Example:** `Router# set memory debug incremental starting-time`	Sets the starting time for incremental analysis to the time when the command is issued.
Step 3	**`show memory debug incremental allocations`** or **`show memory debug incremental leaks`** or **`show memory debug incremental leaks lowmem`** or **`show memory debug incremental status`** **Example:** `Router# show memory debug incremental allocations` or **Example:** `Router# show memory debug incremental leaks` or **Example:** `Router# show memory debug incremental leaks lowmem` or **Example:** `Router# show memory debug incremental status`	Displays all the memory blocks that were allocated after the issue of a **set memory debug incremental starting-time** command. The displayed memory blocks are just memory allocations, they are not necessarily leaks. or Displays output similar to the **show memory debug leaks** command, except that it displays only memory that was leaked after the issue of a **set memory debug incremental starting-time** command. or Forces memory leak detection to work in low memory mode. The output for this command is similar to the **show memory debug leaks** command, except that it displays only memory that was leaked after the issue of a **set memory debug incremental starting-time** command. • In low memory mode, the analysis time is considerably greater than it is in normal mode. • You can use this command when you already know that normal mode memory leak detection will fail (perhaps by an unsuccessful previous attempt to invoke normal mode memory leak detection). or Displays whether a starting point for incremental analysis has been set and the elapsed time since then.

Examples

This section provides the following output examples:

- Sample Output for the show memory debug incremental allocations Command, page 10
- Sample Output for the show memory debug incremental status Command, page 10

Sample Output for the show memory debug incremental allocations Command

The following example shows output from the **show memory debug incremental** command when entered with the **allocations** keyword:

```
Router# show memory debug incremental allocations

Address     Size   Alloc_pc  PID  Name
62DA4E98     176 608CDC7C  44   CDP Protocol
62DA4F48      88 608CCCC8  44   CDP Protocol
62DA4FA0      88 606224A0  3    Exec
62DA4FF8      96 606224A0  3    Exec
635BF040      96 606224A0  3    Exec
63905E50     200 606A4DA4  69   Process Events
```

Sample Output for the show memory debug incremental status Command

The following example shows output from the **show memory debug incremental** command entered with the **status** keyword:

```
Router# show memory debug incremental status

Incremental debugging is enabled
Time elapsed since start of incremental debugging: 00:00:10
```

Additional References

The following sections provide references related to Memory Leak Detector.

Related Documents

Related Topic	Document Title
Additional commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*The Cisco IOS Configuration Fundamentals and Network Management Command Reference*, Release 12.3 T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Configuration Fundamentals Command Reference* at http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **set memory debug incremental starting-time**
- **show memory debug incremental**
- **show memory debug leaks**

Rebooting

Rebooting and Reloading - Configuring Image Loading Characteristics

This chapter describes the basic procedure a Cisco device (such as a router) performs when it reboots, how to alter the procedure, and how to use the ROM monitor.

For a complete description of the booting commands mentioned in this chapter, refer to the "Booting Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Understanding Rebooting Procedures

The following sections describe what happens when the router reboots:

- Which Configuration File Does the Router Use upon Startup?
- Which Image Does the Router Use upon Startup?

Which Configuration File Does the Router Use upon Startup?

On all platforms except Class A Flash file system platforms:

- If the configuration register is set to ignore NVRAM, the router enters setup mode.
- If the configuration register is not set to ignore NVRAM,
  - The startup software checks for configuration information in NVRAM.
  - If NVRAM holds valid configuration commands, the Cisco IOS software executes the commands automatically at startup.
  - If the software detects a problem with NVRAM or the configuration it contains (a CRC checksum error), it enters **setup** mode and prompts for configuration.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

On Class A Flash file system platforms:

- If the configuration register is set to ignore NVRAM, the router enters setup mode.
- If the configuration register is not set to ignore NVRAM,
  - The startup software uses the configuration pointed to by the CONFIG_FILE environment variable.
  - When the CONFIG_FILE environment variable does not exist or is null (such as at first-time startup), the router uses NVRAM as the default startup device.
  - When the router uses NVRAM to start up and the system detects a problem with NVRAM or the configuration it contains, the router enters **setup** mode.

Problems can include a bad checksum for the information in NVRAM or an empty NVRAM with no configuration information. Refer to the "Troubleshooting Hardware and Booting Problems" chapter publication *Internetwork Troubleshooting Guide* for troubleshooting procedures. See the "Using Setup for Configuration Changes" chapter in this publication for details on the **setup** command facility. For more information on environment variables, refer to the "Setting Environment Variables" section.

Which Image Does the Router Use upon Startup?

When a router is powered on or rebooted, the following events happen:

- The ROM monitor initializes.
- The ROM monitor checks the boot field (the lowest four bits) in the configuration register.
  - If the last digit of the boot field is 0 (for example, 0x100), the system does not boot. Instead the system enters ROM monitor mode and waits for user intervention. From ROM monitor mode, you can manually boot the system using the **boot** or **b** command.
  - If the last digit of the boot field is1 (for example, 0x101), the boot helper image is loaded from ROM. (On some platforms, the boot helper image is specified by the BOOTLDR environment variable.)
  - If the last digit of the boot field is 2 through F (for example, 0x102 through 0x10F), the router boots the first valid image specified in the configuration file or specified by the BOOT environment variable.

Note The configuration register boot field value is expressed in hexadecimal. Because the boot field only encompasses the last four bits (represented by the last hexadecimal digit) of the configuration register value, the only digit we are concerned with in this discussion is the last digit. The makes 0x1 (0000 0001) equivalent to 0x101 (1 0000 0001) in discussions of the boot field, as in both cases the last four bits are 0001.

When the boot field is 0x102 through 0x10F, the router goes through each **boot system** command in order until it boots a valid image. If bit 13 in the configuration register is set, each command will be tried once (bit 13 is indicated by the position occupied by *b* in the following hexadecimal notation: 0x*b*000). If bit 13 is not set, the **boot system** commands specifying a network server will be tried up to five more times. The timeouts between each consecutive attempt are 2, 4, 16, 256, and 300 seconds.

If the router cannot find a valid image, the following events happen:

- If all boot commands in the system configuration file specify booting from a network server and all commands fail, the system attempts to boot the first valid file in Flash memory.

- If the "boot-default-ROM-software" option in the configuration register is set, the router will start the boot image (the image contained in boot ROM or specified by the BOORLDR environment variable).
- If the "boot-default-ROM-software" option in the configuration register is not set, the system waits for user intervention at the ROM monitor prompt. You must boot the router manually.
- If a fully functional system image is not found, the router will not function and must be reconfigured through a direct console port connection.

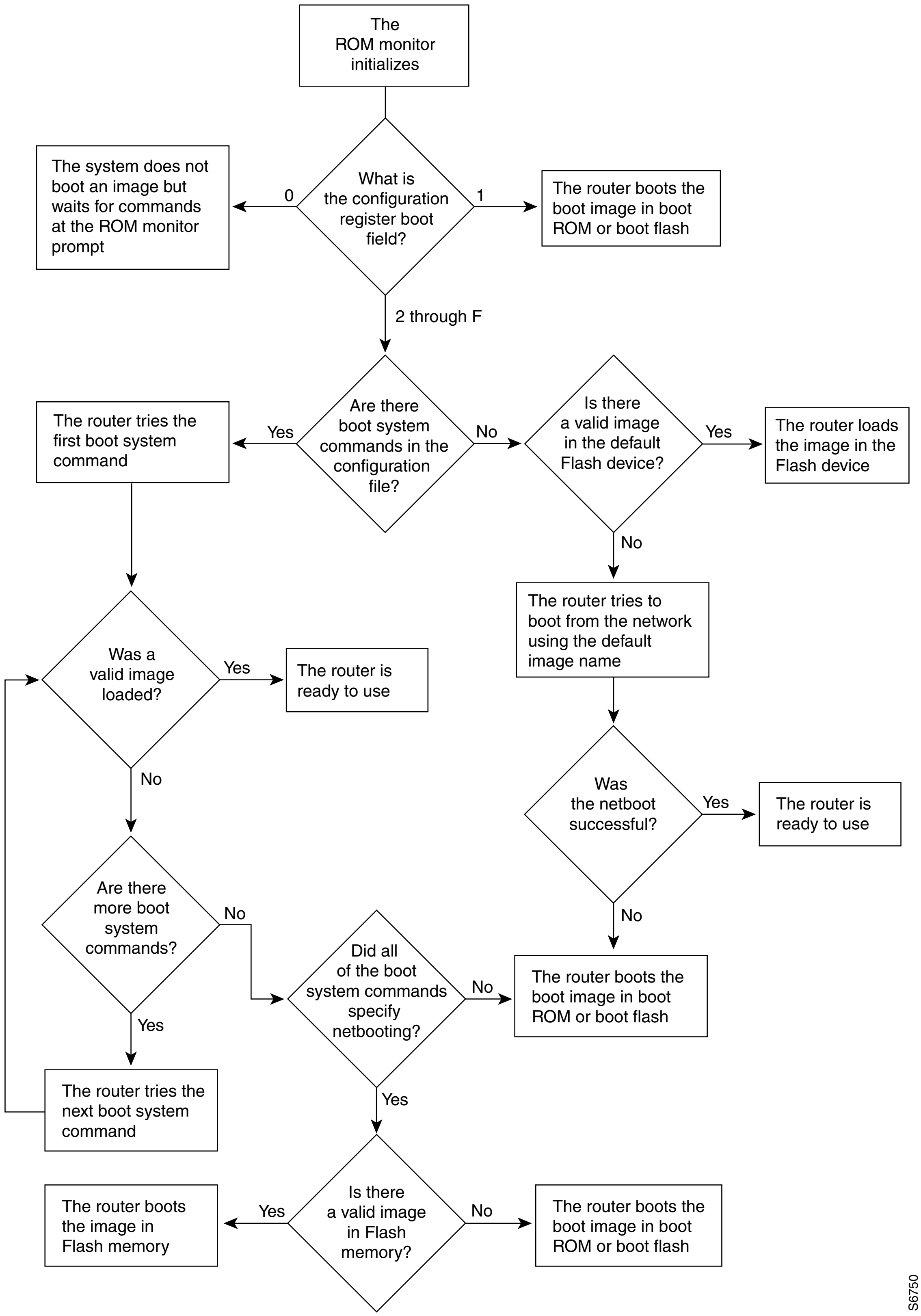
Note Refer to your platform documentation for information on the default location of the boot image.

When looking for a bootable file in Flash memory:

- The system searches for the filename in Flash memory. If a filename is not specified, the software searches through the entire Flash directory for a bootable file instead of picking only the first file.
- The system attempts to recognize the file in Flash memory. If the file is recognized, the software decides whether it is bootable by performing the following checks:
  - For run-from-Flash images, the software determines whether it is loaded at the correct execution address.
  - For run-from-RAM images, the software determines whether the system has enough RAM to execute the image.

Figure 12 illustrates the basic booting decision process.

Figure 12 Booting Process



Rebooting Task List

Tasks related to rebooting are described in the following sections:

- Displaying Boot Information
- Modifying the Configuration Register Boot Field
- Setting Environment Variables
- Scheduling a Reload of the System Image
- Entering ROM Monitor Mode
- Manually Loading a System Image from ROM Monitor

Displaying Boot Information

Use the following commands in EXEC mode to display information about system software, system image files, and configuration files:

Command	Purpose
`Router#` **`show bootvar`**	Lists the contents of the BOOT environment variable, the name of the configuration file pointed to by the CONFIG_FILE environment variable, and the contents of the BOOTLDR environment variable.
`Router#` **`more nvram:startup-config`**	Lists the startup configuration information. On all platforms except the Class A Flash file systems, the startup configuration is usually in NVRAM. On Class A Flash file systems, the CONFIG_FILE environment variable points to the startup configuration, defaulting to NVRAM.
`Router#` **`show version`**	Lists the system software release version, system image name, configuration register setting, and other information.

Refer to the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference* for examples of these commands.

You can also use the o command (or the **confreg** command for some platforms) in ROM monitor mode to list the configuration register settings on some platforms.

Modifying the Configuration Register Boot Field

The configuration register boot field determines whether the router loads an operating system image, and if so, where it obtains this system image. This section contains the following topics:

- How the Router Uses the Boot Field
- Hardware Versus Software Configuration Register Boot Fields
- Modifying the Software Configuration Register Boot Field

Refer to the documentation for your platform for more information on the configuration register.

How the Router Uses the Boot Field

The lowest four bits of the 16-bit configuration register (bits 3, 2, 1, and 0) form the boot field. The following boot field values determine if the router loads an operating system and where it obtains the system image:

- When the entire boot field equals 0-0-0-0 (0x0), the router does not load a system image. Instead, it enters ROM monitor or "maintenance" mode from which you can enter ROM monitor commands to manually load a system image. Refer to the "Manually Loading a System Image from ROM Monitor" section for details on ROM monitor mode.
- When the entire boot field equals 0-0-0-1 (0x1), the router loads the boot helper or rxboot image.
- When the entire boot field equals a value between 0-0-1-0 (0x2) and 1-1-1-1 (0xF), the router loads the system image specified by **boot system** commands in the startup configuration file. When the startup configuration file does not contain **boot system** commands, the router tries to load a default system image stored on a network server.

  When loading a default system image from a network server, the router uses the configuration register settings to determine the default system image filename for booting from a network server. The router forms the default boot filename by starting with the word cisco and then appending the octal equivalent of the boot field number in the configuration register, followed by a hyphen (-) and the processor type name (cisco*nn-cpu*). See the appropriate hardware installation guide for details on the configuration register and the default filename.

Hardware Versus Software Configuration Register Boot Fields

You modify the boot field from either the hardware configuration register or the software configuration register, depending on the platform.

Most platforms have use a software configuration register. Refer to your hardware documentation for information on the configuration register for your platform.

The hardware configuration register can be changed only on the processor card with dual in-line package (DIP) switches located at the back of the router. For information on modifying the hardware configuration register, refer to the appropriate hardware installation guide.

Modifying the Software Configuration Register Boot Field

To modify the software configuration register boot field, use the following commands:

	Command	Purpose
Step 1	`Router#` **`show version`**	Obtains the current configuration register setting. The configuration register is listed as a hexadecimal value.
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`config-register`** *`value`*	Modifies the existing configuration register setting to reflect the way in which you want to load a system image. The configuration register value is in hexadecimal form with a leading "0x."
Step 4	`Router(config)#` **`end`**	Exits configuration mode.

	Command	Purpose
Step 5	`Router# show version`	(Optional) Verifies that the configuration register setting is correct. Repeat steps 2 through 5 if the setting is not correct.
Step 6	`Router# copy running-config startup-config`	Saves the running configuration to the startup configuration.
Step 7	`Router# reload`	(Optional) Reboots the router to make your changes take effect.

In ROM monitor mode, use the **o** command or the **confreg** command on some platforms to list the value of the configuration register boot field.

Modify the current configuration register setting to reflect the way in which you want to load a system image. To do so, change the least significant hexadecimal digit to one of the following:

- 0 to load the system image manually using the **boot** command in ROM monitor mode.
- 1 to load the system image from boot ROMs. On the Cisco 7200 series and Cisco 7500 series, this setting configures the system to automatically load the system image from bootflash.
- 2–F to load the system image from **boot system** commands in the startup configuration file or from a default system image stored on a network server.

For example, if the current configuration register setting is 0x101 and you want to load a system image from **boot system** commands in the startup configuration file, you would change the configuration register setting to 0x102.

Modifying the Software Configuration Register Boot Field Example

In the following example, the **show version** command indicates that the current configuration register is set so that the router does not automatically load an operating system image. Instead, it enters ROM monitor mode and waits for user-entered ROM monitor commands. The new setting instructs the router to a load a system image from commands in the startup configuration file or from a default system image stored on a network server.

```
Router1# show version

Cisco IOS (tm) Software
4500 Software (C4500-J-M), Version 11.1(10.4), RELEASE SOFTWARE
Copyright (c) 1986-1997 by Cisco Systems, Inc.
Compiled Mon 07-Apr-97 19:51 by lmiller
Image text-base: 0x600088A0, data-base: 0x60718000

ROM: System Bootstrap, Version 5.1(1), RELEASE SOFTWARE (fc1)
FLASH: 4500-XBOOT Bootstrap Software, Version 10.1(1), RELEASE SOFTWARE (fc1)

Router1 uptime is 6 weeks, 5 days, 2 hours, 22 minutes
System restarted by error - a SegV exception, PC 0x6070F7AC
System image file is "c4500-j-mz.111-current", booted via flash

cisco 4500 (R4K) processor (revision 0x00) with 32768K/4096K bytes of memory.
Processor board ID 01242622
R4600 processor, Implementation 32, Revision 1.0
G.703/E1 software, Version 1.0.
Bridging software.
SuperLAT software copyright 1990 by Meridian Technology Corp).
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
TN3270 Emulation software (copyright 1994 by TGV Inc).
Basic Rate ISDN software, Version 1.0.
2 Ethernet/IEEE 802.3 interfaces.
2 Token Ring/IEEE 802.5 interfaces.
4 ISDN Basic Rate interfaces.
```

```
128K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read/Write)
4096K bytes of processor board Boot flash (Read/Write)

Configuration register is 0x2100

Router1# configure terminal
Router1(config)# config-register 0x210F
Router1(config)# end
Router1# reload
```

Setting Environment Variables

Because many platforms can boot images from several locations, these systems use special ROM monitor environment variables to specify the location and filename of images that the router is to use. In addition, Class A Flash file systems can load configuration files from several locations and use an environment variable to specify startup configurations.

These special environment variables are as follows:

- BOOT Environment Variable
- BOOTLDR Environment Variable
- CONFIG_FILE Environment Variable

BOOT Environment Variable

The BOOT environment variable specifies a list of bootable system images on various file systems. Refer to the “Specify the Startup System Image in the Configuration File” section in the “Loading and Maintaining System Images and Microcode” chapter of the *Configuration Fundamentals Configuration Guide*. After you save the BOOT environment variable to your startup configuration, the router checks the variable upon startup to determine the device and filename of the image to boot.

The router tries to boot the first image in the BOOT environment variable list. If the router is unsuccessful at booting that image, it tries to boot the next image specified in the list. The router tries each image in the list until it successfully boots. If the router cannot boot any image in the BOOT environment variable list, the router attempts to boot the boot image.

If an entry in the BOOT environment variable list does not specify a device, the router assumes the device is **tftp**. If an entry in the BOOT environment variable list specifies an invalid device, the router skips that entry.

BOOTLDR Environment Variable

The BOOTLDR environment specifies the Flash file system and filename containing the boot image that the ROM monitor uses if it cannot find a valid system image. In addition, a boot image is required to boot the router with an image from a network server.

You can change the BOOTLDR environment variable on platforms that use a software boot image rather than boot ROMs. On these platforms, the boot image can be changed without having to replace the boot ROM.

This environment variable allows you to have several boot images. After you save the BOOTLDR environment variable to your startup configuration, the router checks the variable upon startup to determine which boot image to use if the system cannot be loaded.

Note Refer to your platform documentation for information on the default location of the boot image.

CONFIG_FILE Environment Variable

For Class A Flash file systems, the CONFIG_FILE environment variable specifies the file system and filename of the configuration file to use for initialization (startup). Valid file systems can include **nvram:**, **bootflash:**, **slot0:**, and **slot1:**. Refer to the "Location of Configuration Files" section on page 2 in the "Modifying, Downloading, and Maintaining Configuration Files" chapter for more information on devices. After you save the CONFIG_FILE environment variable to your startup configuration, the router checks the variable upon startup to determine the location and filename of the configuration file to use for initialization.

The router uses the NVRAM configuration during initialization when the CONFIG_FILE environment variable does not exist or when it is null (such as at first-time startup). If the router detects a problem with NVRAM or a checksum error, the router enters **setup** mode. Refer to the "Using Setup for Configuration Changes" chapter in this publication for more information on the **setup** command facility.

Controlling Environment Variables

Although the ROM monitor controls environment variables, you can create, modify, or view them with certain commands. To create or modify the BOOT, BOOTLDR, and CONFIG_FILE environment variables, use the **boot system**, **boot bootldr**, and **boot config** global configuration commands, respectively.

Refer to the "Specify the Startup System Image in the Configuration File" section in the "Loading and Maintaining System Images" chapter of this book for details on setting the BOOT environment variable. Refer to the "Specify the Startup Configuration File" section in the "Managing Configuration Files" chapter of this document for details on setting the CONFIG_FILE variable.

Note When you use these three global configuration commands, you affect only the running configuration. You must save the environment variable settings to your startup configuration to place the information under ROM monitor control and for the environment variables to function as expected. Use the **copy system:running-config nvram:startup-config** command to save the environment variables from your running configuration to your startup configuration.

You can view the contents of the BOOT, BOOTLDR, and the CONFIG_FILE environment variables by issuing the **show bootvar** command. This command displays the settings for these variables as they exist in the startup configuration as well as in the running configuration if a running configuration setting differs from a startup configuration setting.

Use the **more nvram:startup-config** command to display the contents of the configuration file pointed to by the CONFIG_FILE environment variable.

Setting the BOOTLDR Environment Variable

To set the BOOTLDR environment variable, use the following commands, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# dir [flash-filesystem:]`	Verifies that internal Flash or bootflash contains the boot helper image.
Step 2	`Router# configure terminal`	Enters the configuration mode from the terminal.
Step 3	`Router(config)# boot bootldr file-url`	Sets the BOOTLDR environment variable to specify the Flash device and filename of the boot helper image. This step modifies the runtime BOOTLDR environment variable.
Step 4	`Router# end`	Exits configuration mode.
Step 5	`Router# copy system:running-config nvram:startup-config`	Saves the configuration you just performed to the system startup configuration.
Step 6	`Router# show bootvar`	(Optional) Verifies the contents of the BOOTLDR environment variable.

The following example sets the BOOTLDR environment to change the location of the boot helper image from internal Flash to slot 0.

```
Router# dir bootflash:
-#- -length- -----date/time------ name
1   620      May 04 1995 26:22:04 rsp-boot-m
2   620      May 24 1995 21:38:14 config2

7993896 bytes available (1496 bytes used)
Router# configure terminal
Router (config)# boot bootldr slot0:rsp-boot-m
Router (config)# end
Router# copy system:running-config nvram:startup-config
[ok]
Router# show bootvar
BOOT variable = slot0:rsp-boot-m
CONFIG_FILE variable = nvram:
Current CONFIG_FILE variable = slot0:router-config

Configuration register is 0x0
```

Scheduling a Reload of the System Image

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

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

Configuring a Scheduled Reload

To configure the router to reload the Cisco IOS software at a later time, use one of the following commands in privileged EXEC command mode:

Command	Purpose
`Router# reload in [hh:]mm [text]`	Schedules a reload of the software to take effect in *mm* minutes (or *hh* hours and *mm* minutes) from now.
`Router# reload at hh:mm [month day | day month] [text]`	Schedules a 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.

Note The **at** keyword can only be used if the system clock has been set on the router (either through NTP, the hardware calendar, or manually). The time is relative to the configured time zone on the router. To schedule reloads across several routers to occur simultaneously, the time on each router must be synchronized with NTP. For information on configuring NTP, see the "Performing Basic System Management" chapter in the *Cisco IOS Network Management Configuration Guide*, Release 12.4.

The following example illustrates how to use the **reload** command to reload the software on the router on the current day at 7:30 p.m.:

```
Router# reload at 19:30
Reload scheduled for 19:30:00 UTC Wed Jun 5 1996 (in 2 hours and 25 minutes)
Proceed with reload? [confirm]
```

The following example illustrates how to use the **reload** command to reload the software on the router at a future time:

```
Router# reload at 02:00 jun 20
Reload scheduled for 02:00:00 UTC Thu Jun 20 1996 (in 344 hours and 53 minutes)
Proceed with reload? [confirm]
```

Display Information about a Scheduled Reload

To display information about a previously scheduled reload or to determine if a reload has been scheduled on the router, use the following command in EXEC command mode:

Command	Purpose
`Router# show reload`	Displays reload information, including the time the reload is scheduled to occur, and the reason for the reload if it was specified when the reload was scheduled.

Cancel a Scheduled Reload

To cancel a previously scheduled reload, use the following command in privileged EXEC command mode:

Command	Purpose
Router# **reload cancel**	Cancels a previously scheduled reload of the software.

The following example illustrates how to use the **reload cancel** command to stop a scheduled reload:

```
Router# reload cancel
Router#
***
*** --- SHUTDOWN ABORTED ---
***
```

Entering ROM Monitor Mode

During the first 60 seconds of startup, you can force the router to stop booting. The router will enter ROM monitor mode, where you can change the configuration register value or boot the router manually.

To stop booting and enter ROM monitor mode, use the following commands in EXEC mode:

	Command	Purpose
Step 1	Router# **reload** Press the Break[1] key during the first 60 seconds while the system is booting.	Enter ROM monitor mode from privileged EXEC mode.
Step 2	**?**	List the ROM monitor commands.

1. This key will not work on the Cisco 7000 unless it has at least Cisco IOS Release 10 boot ROMs.

Timesaver If you are planning to use ROM monitor mode on a regular basis, or wish users to load using ROM monitor commands, you can configure the system to default to ROMMON. To automatically boot your system in ROM monitor mode, reset the configuration register to 0x0 by using the **config-register 0x0** configuration command. The new configuration register value, 0x0, takes effect after the router or access server is rebooted with the **reload** command. If you set the configuration to 0x0, you will have to manually boot the system from the console each time you reload the router or access server.

To exit ROMMON mode, use the continue command. If you have changed the configuration, use the **copy running-config startup-config** command and then issue the **reload** command to save your configuration changes.

Aliasing ROM Monitoring Commands

The ROM monitor supports command aliasing modeled on the aliasing function built into the Korn shell. The **alias** command is used to set and view aliased names. This allows the user to alias command names to a letter or word. Aliasing is often used to shorten command names or automatically invoke command options.

Aliases are stored in NVRAM and remain intact across periods of no power. These are some of the set aliases:

- **b**—boot
- **h**—history
- **i**—intialize/reset
- **r**—repeat
- **k**—stack
- **?**—help

The following example shows a pre-aliased menu-type list for ROMMON commands:

```
> ?
$ state      Toggle cache state (? for help)
B [filename] [TFTP Server IP address | TFTP Server Name]
             Load and execute system image from ROM or from TFTP server
C [address]  Continue execution [optional address]
D /S M L V   Deposit value V of size S into location L with modifier M
E /S M L     Examine location L with size S with modifier M
G [address]  Begin execution
H            Help for commands
I            Initialize
K            Stack trace
L [filename] [TFTP Server IP address | TFTP Server Name]
             Load system image from ROM or from TFTP server, but do not
             begin execution
O            Show configuration register option settings
P            Set the break point
S            Single step next instruction
T function   Test device (? for help)
Deposit and Examine sizes may be B (byte), L (long) or S (short).
Modifiers may be R (register) or S (byte swap).
Register names are: D0-D7, A0-A6, SS, US, SR, and PC
```

If your options appear in the above menu-type format, you can use the listed aliased commands. To initialize the router or access server, enter the **i** command. The **i** command causes the bootstrap program to reinitialize the hardware, clear the contents of memory, and boot the system. To boot the system image file, use the **b** command.

The ROM monitor software characteristics will vary depending on your platform. For further details on ROM monitor mode commands, refer to the appropriate hardware installation guide, or perform a search on Cisco.com.

Manually Loading a System Image from ROM Monitor

If your router does not find a valid system image, or if its configuration file is corrupted at startup, or the configuration register is set to enter ROM monitor mode, the system enters ROM monitor mode. From this mode, you can manually load a system image from the following locations:

- Internal Flash memory or a Flash memory PC card
- A network server file
- ROM
- A local or remote computer, using the Xmodem or Ymodem protocol (Cisco 1600 series and Cisco 3600 series routers only)

You may only boot from a location if the router can store an image there. Therefore, not all platforms can manually load from these locations.

You can also enter ROM monitor mode by restarting the router and then pressing the **Break** key or issuing a "send break" command from a telnet session during the first 60 seconds of startup.

Manually Booting from Flash Memory in ROMMON

To manually boot from Flash memory, use the following command in ROM monitor mode:

Command	Purpose
`ROMMON >` **`boot flash`** `[`*`filename`*`]` `ROMMON >` **`boot flash`** *`partition-number`***`:`**`[`*`filename`*`]` `ROMMON >` **`boot flash flash:`**`[` *`partition-number`***`:`**`] [`*`filename`*`]` `ROMMON >` **`boot`** `[`*`flash-fs`***`:`**`][`*`partition-number`***`:`**`][`*`filename`*`]` `(Cisco 1600 series and Cisco 3600 series)` `ROMMON >` **`boot`** *`device`***`:`**`[`*`filename`*`] (Cisco 7000 family)`	Manually boot the router from Flash. Refer to your hardware documentation for the correct form of this command to use.

If the filename is not specified, the first bootable file found in the device and partition is used.

In the following example, a router is manually booted from Flash memory. Because the optional *filename* argument is absent, the first valid file in Flash memory is loaded.

```
> boot flash
F3: 1858656+45204+166896 at 0x1000

Booting gs7-k from flash memory RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR [OK -
1903912/13765276 bytes]
F3: 1858676+45204+166896 at 0x1000

              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
```

In the following example, the **boot flash** command is used with the filename gs7-k—the name of the file that is loaded:

```
> boot flash gs7-k
F3: 1858656+45204+166896 at 0x1000

Booting gs7-k from flash memory RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
```

```
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRR [OK - 1903912/13765276 bytes]
F3: 1858676+45204+166896 at 0x1000

              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
System Bootstrap, Version 4.6(1012) [mlw 99], INTERIM SOFTWARE
Copyright (c) 1986-1992 by cisco Systems
RP1 processor with 16384 Kbytes of memory
```

The following command instructs the ROM monitor to boot the first file in the first partition of internal Flash memory:

```
> boot flash:
```

This command instructs the ROM monitor to boot the first file in the second partition of the Flash memory card in slot 0:

```
> boot slot0:2:
```

In this example, the ROM monitor boots the file named image name from the third partition of the Flash memory card in slot 0:

```
> boot slot0:3:imagename
```

The following command fails to specify a valid device type (**flash:**, **slot0:**, or **slot1:**), so the ROM monitor invokes the boot helper to boot a system image.

```
> boot flash
```

Manually Booting from a Network File in ROMMON

To manually boot from a network file, use the following command in ROM monitor mode:

Command	Purpose
ROMMON > **boot** *filename* [*ip-address*]	Manually boots the router from a network file.

In the following example, a router is manually booted from the network file *network1*:

```
>boot network1
```

Manually Booting from ROM in ROMMON

To manually boot the router from ROM, use the following command in ROM monitor mode:

Command	Purpose
ROMMON > **boot**	Manually boots the router from ROM.

On the Cisco 7200 series and Cisco 7500 series, the **boot** command loads the first bootable image located in bootflash.

In the following example, a router is manually booted from ROM:

```
>boot
```

Manually Booting Using MOP in ROMMON

You can interactively boot system software using MOP. Typically, you do this to verify that system software has been properly installed on the MOP boot server before configuring the router to automatically boot the system software image.

To manually boot the router using MOP, use the following command in ROM monitor mode:

Command	Purpose
ROMMON > **boot system mop** *filename* [*mac-address*] [*interface*]	Manually boots the router using MOP.

The Cisco 7200 series and Cisco 7500 series do not support the **boot mop** command.

In the following example, a router is manually booted from a MOP server:

```
>boot mop network1
```

Exiting from ROMMON

To return to EXEC mode from the ROM monitor, you must continue loading from the default system image. To exit ROMMON mode and resume loading, use the following command in ROM monitor mode:

Command	Purpose
ROMMON > **continue**	Resumes loading the startup configuration file and brings the user to EXEC mode.