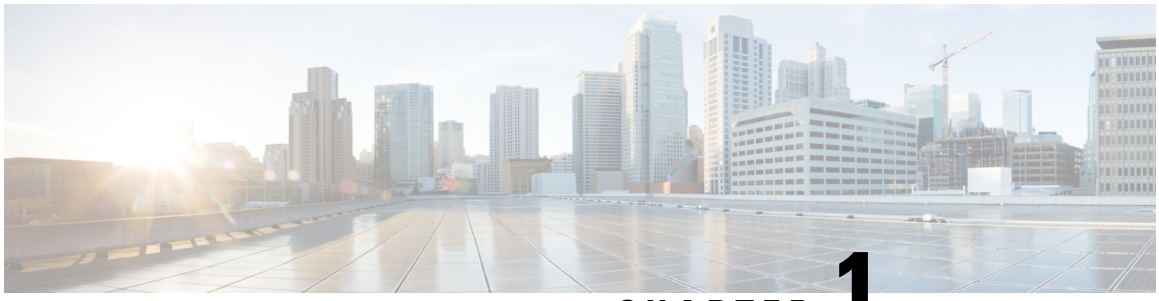
The Integrated File System Configuration Guide, Cisco IOS XE Fuji 16.9.x

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

CONTENTS



CHAPTER 1

Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E (for Catalyst Switching) and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

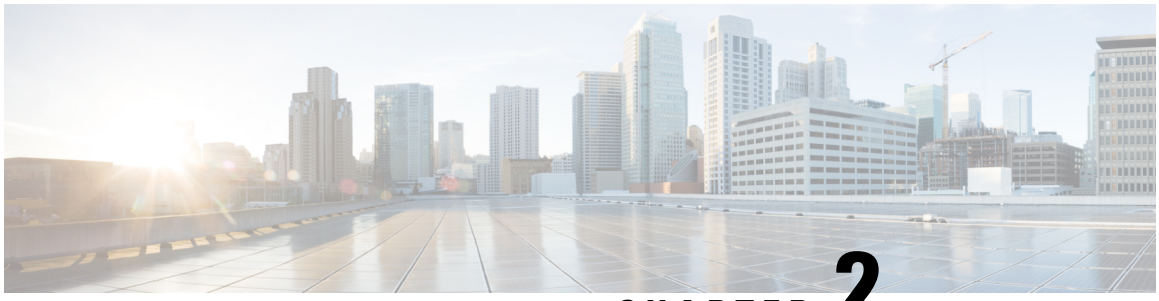
Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the . RSS feeds are a free service.



CHAPTER 2

Configuring Basic File Transfer Services

Using basic file transfer services, you can 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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Basic File Transfer Services

- You should have at least a basic familiarity with the Cisco IOS environment and the command-line interface.
- You should have at least a minimal configuration running on your system.

Restrictions for Basic File Transfer Services

- You must have your network up and running, with Cisco IOS Release 12.2 or a later release installed.
- Some of the Cisco IOS configuration commands are only available on certain router platforms, and the command syntax may vary on different platforms.

Information About Basic File Transfer Services

Use of 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.

Use of 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.

Use of 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.

Use of a Router for 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.

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. 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.

Implementation of 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.

Implementation of 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.

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.

Configure the Remote Client 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 **iprcmdremote-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.

1. 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 **iprcmdremote-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 personalcomputer used as a file server, the computer must support rsh.

Use of a Router for FTP Connections

You can 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

How to Configure Basic File Transfer Services

Configuring the Router for Use as a TFTP Server

To configure your router for use as a TFTP server, complete the tasks in this section.

Before you begin

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, andping 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **tftp-server flash** [*partition-number***:**]*filename1* [**alias***filename2*] [*access-list-number*]
   - **tftp-server flash** *device* **:** *filename* (Cisco 7000 family only)
   - **tftp-server flash** [*device***:**][*partition-number***:**]*filename* (Cisco 1600 series and Cisco 3600 series only)
   - **tftp-server rom alias** *filename1* [*access-list-number*]
4. **end**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following: • **tftp-server flash** [*partition-number***:**]*filename1* [**alias***filename2*] [*access-list-number*] • **tftp-server flash** *device* **:** *filename* (Cisco 7000 family only) • **tftp-server flash** [*device***:**][*partition-number***:**]*filename* (Cisco 1600 series and Cisco 3600 series only) • **tftp-server rom alias** *filename1* [*access-list-number*] **Example:** `Device(config)# tftp-server flash version-10.3 22`	Specifies the system image to send in response to Read Requests. You can enter multiple lines to specify multiple images.
Step 4	**end** **Example:** `Device(config)# end`	Ends the configuration session and returns you to privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:**	Saves the running configuration to the startup configuration file.

	Command or Action	Purpose
	`Device# copy running-config startup-config`	

Examples

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
```

In the following example, a router sends 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#
```

Troubleshooting

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.

Configuring the Client Router

To configure the client router to first load a system image from the server, and as a backup, to configure the client router to load its own ROM image if the load from a server fails, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no boot system**
4. **boot system** [**tftp**] *filename* [*ip-address*]
5. **boot system rom**
6. **config-register** *value*
7. **end**
8. **copy running-config startup-config**
9. **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no boot system** **Example:** `Device(config)# no boot system`	(Optional) Removes all previous **bootsystem** statements from the configuration file.
Step 4	**boot system** [**tftp**] *filename* [*ip-address*] **Example:** `Device(config)# boot system c5300-js-mz.121-5.T.bin 172.16.1.1`	Specifies that the client router load a system image from the server.
Step 5	**boot system rom** **Example:** `Device(config)# boot system rom`	Specifies that the client router loads its own ROM image if the load from a server fails.
Step 6	**config-register** *value* **Example:** `Device(config)# config-register 0x010F`	Sets the configuration register to enable the client router to load a system image from a network server.
Step 7	**end** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`Device(config)# end`	
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the configuration file to your startup configuration.
Step 9	**reload** **Example:** `Device# reload`	(Optional) Reloads the router to make your changes take effect.

Examples

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 172.16.1.1

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 **nobootsystem** command invalidates all other**bootsystem** commands currently in the configuration memory, and any **bootsystem** commands entered after this command will be executed first. The second command, **bootsystem***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 **bootsystemrom** command, which is included as a backup in case of a network problem. The **copyrunning-configstartup-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 **showversion** command after the router has rebooted:

```
Device> 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.

What to Do Next

After the system reloads, you should use the **showversion** EXEC mode command to verify that the system booted the desired image.

Caution Using the **nobootsystem** 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.

Configuring the Router as a RARP Server

To configure the router as a RARP server, complete the tasks in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type* [*slot*/]*port*
4. **ip rarp-server** *ip-address*

DETAILED STEPS

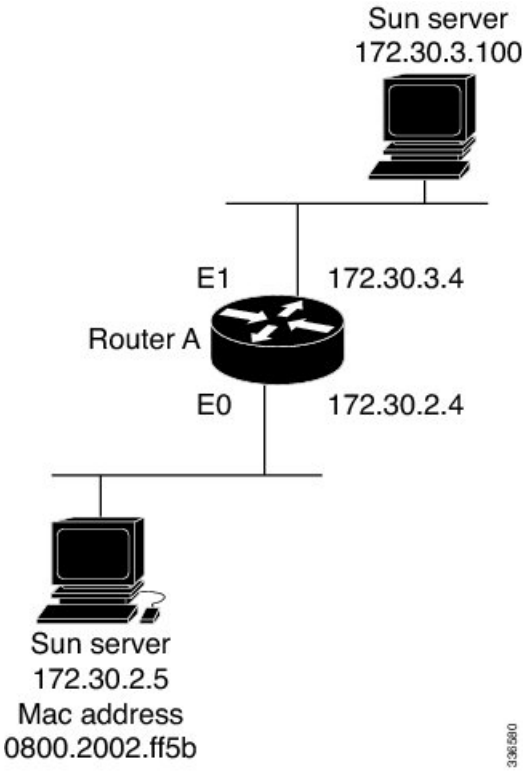
	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type* [*slot*/]*port* **Example:** `Device(config)# interface Gigabitethernet 0/0`	Specifies the interface that you will be configuring the RARP service on and enters interface configuration mode for the specified interface.
Step 4	**ip rarp-server** *ip-address* **Example:** `Device(config-if)# ip rarp-server 172.30.3.100`	Enables the RARP service on the router.

Examples

The figure below 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 1: 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 Gigabitethernet 0/0
! 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 Gigabitethernet 0/0
! 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 a Router to Use rsh and rcp

Specifying the Source Interface for Outgoing RCMD Communications

To configure the router so that RCMD connections use the loopback interface as the source address of all packets leaving the router, specify the interface associated with RCMD communications by completing the task in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rcmd source-interface** *interface-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rcmd source-interface** *interface-id* **Example:** `Device(config)# ip rcmd source-interface`	Specifies the interface address that will be used to label all outgoing rsh and rcp traffic.

Disabling DNS Reverse Lookup for rcmd

DNS Reverse Lookup for rcmd is enabled by default. You can disable the DNS check for RCMD (rsh and rcp) access by completing the task in this section:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip rcmd domain-lookup**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ip rcmd domain-lookup** **Example:** `Device(config)# no ip rcmd domain-lookup`	Disables the Domain Name Service (DNS) reverse lookup function for remote command (rcmp) applications (rsh and rcp).

Configuring the Router to Allow Remote Users to Execute Commands Using rsh

To configure the router to allow remote user to execute commands using rsh, complete the tasks in this section:

SUMMARY STEPS

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

3. **ip rcmd remote-host** *local-username* {*ip-address* | *host* } *remote-username* [**enable**[*level*]]
4. **ip rcmd rsh-enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rcmd remote-host** *local-username* {*ip-address* | *host* } *remote-username* [**enable**[*level*]] **Example:** `Device(config)# ip rcmd remote-host Router1 172.16.101.101 netadmin4 enable`	Creates an entry in the local authentication database for each remote user who is allowed to execute rsh commands.
Step 4	**ip rcmd rsh-enable** **Example:** `Device(config)# ip rcmd rsh-enable`	Enables the software to support incoming rsh commands. **Note** To disable the software from supporting incoming rsh commands , use the **noiprcmdrsh-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.

Examples

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

To execute a command remotely on a network server using rsh, use the following commands in user EXEC mode:

SUMMARY STEPS

1. **enable**
2. **rsh** {*ip-address* | *host* } [/**user***username*] *remote-command*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**rsh** {*ip-address* | *host* } [/**user***username*] *remote-command* **Example:** `Device# rsh mysys.cisco.com /user sharon ls -a`	Executes a command remotely using rsh.

Examples

The following example executes the “ls -a” command in the home directory of the user sharon on mysys.cisco.com using rsh:

```
Device# enable
Device# rsh mysys.cisco.com /user sharon ls -a
.
..
.alias
.cshrc
.emacs
.exrc
.history
.login
.mailrc
.newsrc
.oldnewsrc
.rhosts
.twmrc
.xsession
jazz
Device#
```

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:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rcmd remote-host** *local-username* {*ip-address* | *host* } *remote-username* [**enable**[*level*]]
4. **ip rcmd rcp-enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rcmd remote-host** *local-username* {*ip-address* | *host* } *remote-username* [**enable**[*level*]] **Example:** `Device(config)# ip rcmd remote-host Router1 172.16.101.101 netadmin3`	Create an entry in the local authentication database for each remote user who is allowed to execute rcp commands. **Note** To disable the software from supporting incoming rcp requests, use the **noiprcmdrcp-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.
Step 4	**ip rcmd rcp-enable** **Example:** `Device(config)# ip rcmd rcp-enable`	Enable the software to support incoming rcp requests.

Examples

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

To override the default remote username sent on rcp requests, use the following command in global configuration mode:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rcmd remote-username** *username*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rcmd remote-username** *username* **Example:** `Device(config)# ip rcmd remote-username sharon`	Specifies the remote username. **Note** To remove the remote username and return to the default value, use the **noiprcmdremote-username** command.

Configuring a Router to Use FTP Connections

To configure a router to transfer files between systems on the network using the File Transfer Protocol (FTP), complete the tasks in this section to configure the FTP characteristics:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ftp username** *string*
4. **ip ftp password** [*type*] *password*

5. Do one of the following:
   - **ip ftp passive**
   -
   -
   - **no ip ftp passive**
6. **ip ftp source-interface** *interface*

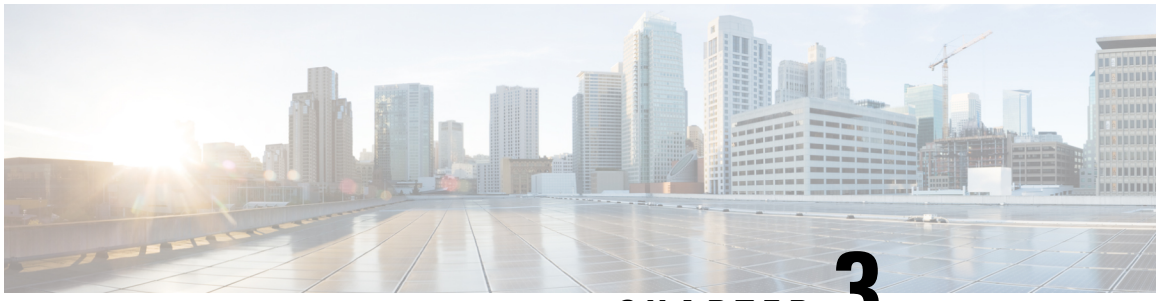
DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip ftp username** *string* **Example:** `Device(config)# ip ftp username zorro`	Specifies the user name to be used for the FTP connection.
Step 4	**ip ftp password** [*type*] *password* **Example:** `Device(config)# ip ftp password sword`	Specifies the password to be used for the FTP connection.
Step 5	Do one of the following: • **ip ftp passive** • • • **no ip ftp passive** **Example:** `Device(config)# ip ftp passive`	Configures the router to only use passive-mode FTP connections. or Allows all types of FTP connections (default).
Step 6	**ip ftp source-interface** *interface* **Example:** `Device(config)# ip ftp source-interface to1`	Specifies the source IP address for FTP connections.

Examples

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 identifies the FTP server
! 192.168.10.3 crashes
exception dump 192.168.10.3
```



CHAPTER 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 HTTP Secure (HTTPS) protocol. HTTP and HTTPS can now be specified as the targets and source locations in Cisco IOS command-line interface (CLI) commands that use file system prefixes such as the **copy** command.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

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.

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.

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.

Note The **copy** command in Cisco IOS Release 12.4T does not work in conjunction with older versions of the Apache server software. The Apache server software must be upgraded to version 2.0.49 or later in order to use the copy command.

Information About File Transfers Using HTTP or HTTPs

To transfer files using HTTP or HTTPS, you should understand the following concept:

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 HTTP Secure transfers, providing secure and authenticated file transfers within the context of a public key infrastructure (PKI).

How to Transfer Files Using HTTP or HTTPs

This section contains the following procedures:

Note 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.

Configuring HTTP Connection Characteristics for File Transfers

Default values are provided for HTTP File transfers. The following task is used to customize the connection characteristics for your network to specify a username and password, connection preferences, a remote proxy server, and the source interface to be used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http client connection** {**forceclose** | **idletimeout***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** | **idletimeout***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 waits 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 from a Remote Server Using HTTP or HTTPs: Example" section for an example.
Step 5	**ip http client password** *password* **Example:**	Specifies the password to be used for HTTP client connections that require user authentication.

	Command or Action	Purpose
	`Router(config)# ip http client password letmein`	**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 from a Remote Server Using HTTP or HTTPs: Example" section for an example.
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` **Example:** `Router#`	Ends your configuration session and returns the CLI to user EXEC mode.

Downloading a File from a Remote Server Using HTTP or HTTPs

Perform this task to download a file from a remote HTTP server using HTTP or HTTPs. The **copy** command helps you to copy any file from a source to a destination.

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - **copy** [**/erase**] [**/noverify**] **http://***remote-source-urllocal-destination-url*
   - **copy https://** *remote-source-url local-destination-url*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable` **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Do one of the following: • **copy** [**/erase**] [**/noverify**] **http://***remote-source-urllocal-destination-url* • **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` **Example:** `Router# copy` **Example:** `copy https://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 **iphttpclientusername** and **iphttpclientpassword** 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.

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 returns error messages to help you determine the cause of a failed copy request. Additional information on the copy process can be displayed with the **debugiphttpclientall** command.

Uploading a File to a Remote Server Using HTTP or HTTPS

Perform this task to upload a file to a remote HTTP server using HTTP or HTTPS.

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - **copy** [**/erase**] [**/noverify**] *local-source-url***http://***remote-destination-url*
   - **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	Do one of the following: • **copy** [**/erase**] [**/noverify**] *local-source-url***http://***remote-destination-url* • **copy** *local-source-url* **https://** *remote-destination-url* **Example:** `Router# http://user1:mypassword@209.165.202.129:8080/image_files/c7200-i-mx_backup` **Example:** `Router# copy flash:c7200-i-mx http://user1:mypassword@209.165.202.129:8080/image_files/c7200-i-mx_backup` **Example:**	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*

	Command or Action	Purpose
		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 **iphttpclientusername** and **iphttpclientpassword** 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.

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 returns error messages to help you determine the cause of a failed copy request. Additional information on the copy process can be displayed with the **debugiphttpclientall** 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**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• 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

Configuring HTTP Connection Characteristics for File Transfers Example

The following example shows how to configure the HTTP password and username for connection to a remote server that authenticates all users. The example also shows how to configure the connection for a 20-second idle connection period. The maximum time the HTTP client waits 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 from a Remote Server Using HTTP or HTTPs Example

The following example shows how to configure 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
```

Uploading a File from Flash to the Remote HTTP Server Example

The following example shows how to copy 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)
```

Downloading a File from the Remote HTTP Server to Flash Memory Example

The following example shows how to copy 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 to a Remote Server Using HTTP or HTTPs

The following example shows how to copy a file to the remote server using HTTP or HTTPs.

```
router#copy flash
: http:
Source filename []? running-config
Address or name of remote host []? 10.1.102.1 Destination filename [pilot-confg]?file1 ...
```

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*
Cisco IOS embedded web server	*HTTP 1.1 Web Server and Client*
Cisco IOS embedded web client	*HTTP 1.1 Client*
Network Management Commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Network Management Command Reference*
Configuration Fundamentals Commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Configuration Fundamentals Command Reference*

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
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
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 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 Transferring Files Using HTTP or HTTPS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 1: Feature Information for Transferring Files Using HTTP or HTTPS

Feature Name	Releases	Feature Information
File Download Using HTTP	12.3(2)T	The File Download Using HTTP feature allows you to copy files from an HTTP server to a Cisco IOS software-based platform.
File Upload Using HTTP	12.3(7)T	
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, and scripts 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.