



Configuring Basic File Transfer Services

Last Updated: May 2, 2008

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.

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.



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

© 2007 Cisco Systems, Inc. All rights reserved.

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.

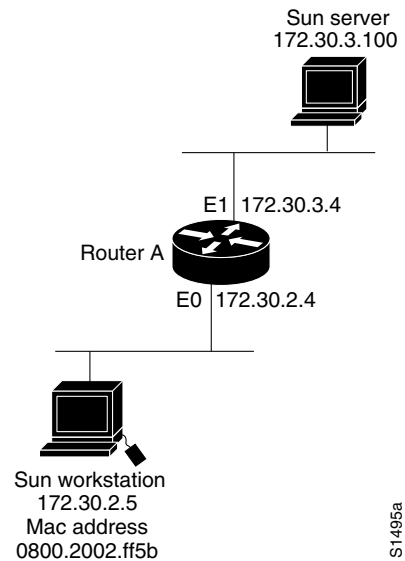
You can configure a Cisco router as a RARP server. This feature enables the Cisco IOS software to answer RARP requests.

To configure the router as a RARP server, use the following commands, beginning in global configuration mode:

Command	Purpose
Router(config)# interface <i>type</i> [<i>slot</i>]/ <i>port</i>	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 <i>ip-address</i>	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 <i>interface-id</i>	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 identify 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 <i>local-username</i> { <i>ip-address</i> <i>host</i> } <i>remote-username</i> [enable [<i>level</i>]]	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 [<i>password</i>]	Enters privileged EXEC mode.
Step 2	Router# rsh { <i>ip-address</i> <i>host</i> } [/user <i>username</i>] <i>remote-command</i>	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 <i>local-username</i> { <i>ip-address</i> <i>host</i> } <i>remote-username</i> [enable [<i>level</i>]]	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 <i>username</i>	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 <i>string</i>	Specifies the user name to be used for the FTP connection.
Router(config)# ip ftp password [<i>type</i>] <i>password</i>	Specifies the password to be used for the FTP connection.
Router(config)# ip ftp passive	Configures the router to only use passive-mode FTP connections.
OR	or
Router(config)# no ip ftp passive	Allows all types of FTP connections (default).
Router(config)# ip ftp source-interface <i>interface</i>	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
```

CCVP, the Cisco logo, and Welcome to the Human Network are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PIX, ProConnect, ScriptShare, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0711R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2008 Cisco Systems, Inc. All rights reserved.

