

# Making Connections to Network Devices

---

This chapter describes how to make remote node, terminal service, and protocol translation connections. It also describes how to monitor and manage these connections. For a complete description of the commands in this chapter, refer to the “Connection Commands” chapter of the *Access Services Command Reference*.

## Connection Service Overview

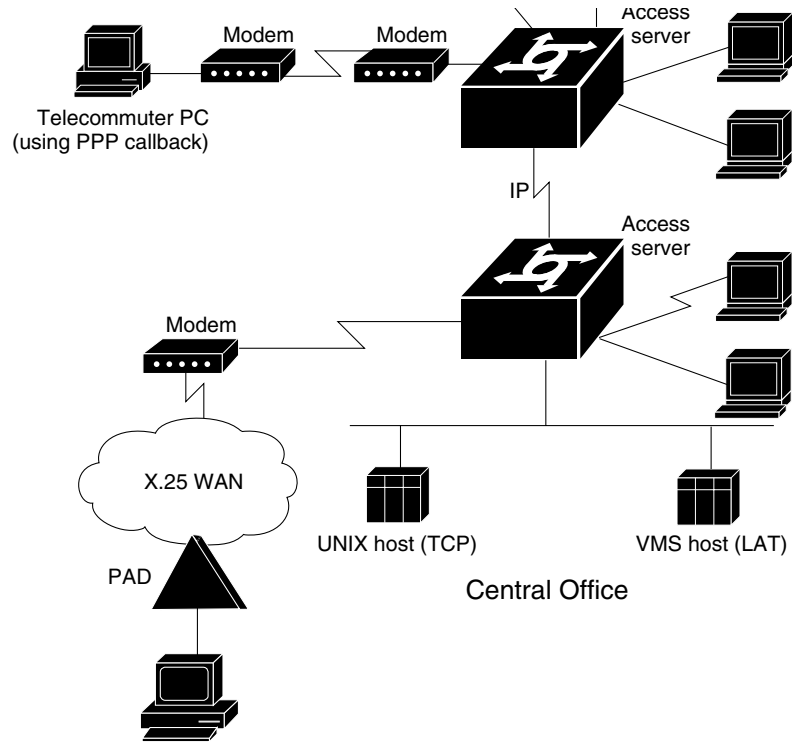
The Cisco IOS software supports the following three types of connection services (see Figure 19):

- Remote node services—Connecting devices over a telephone network using PPP, SLIP, ARA, or XRemote (the NCD X Windows terminal protocol).
  - Point-to-Point Protocol (PPP) and Serial Line Internet Protocol (SLIP) connections—PPP and SLIP provide an inexpensive method for personal computer users to connect across asynchronous dial-up lines into a network to access IP resources (such as printers, file servers, and e-mail). A remote user running PPP or SLIP has the same access to network resources as a person connected directly to the LAN.
  - AppleTalk Remote Access (ARA) connections—ARA is an inexpensive method for Macintosh users to connect across asynchronous dial-up lines into a network to access AppleTalk resources (such as printers, file servers, and e-mail). A remote user running ARA has the same access to network resources as a Macintosh connected directly to the LAN.
  - XRemote—Network Computing Devices (NCD) Inc. XRemote terminal facility supports remote X Windows operation over asynchronous lines.
- Terminal services—Connecting network devices running the same protocol (such as LAT or TCP) across a LAN or WAN through network and terminal-emulation software such as Telnet, rlogin, TN3270, Local Area Transport (LAT), and NetWare Asynchronous Services Interface (NASI).
  - Telnet and rlogin—Of all protocol suites, Transmission Control Protocol/Internet Protocol (TCP/IP) is the most widely implemented on networks of all media types. TCP/IP is today’s standard for internetworking and is supported by most computer vendors, including all UNIX-based workstation manufacturers. TCP/IP includes Telnet and rlogin.
  - NetWare Asynchronous Services Interface (NASI)—Configuring the Cisco IOS software as a NASI server enables NASI clients to connect through your router to network resources.
  - LAT—Local Area Transport (LAT) protocol is Digital Equipment Corporation’s proprietary terminal connection protocol used with Digital minicomputers.
  - TN3270—IBM 3278 terminal emulation provides TN3270-based connectivity to IBM hosts over serial lines.

- Terminal or remote node services using protocol translation—Connecting devices running dissimilar protocols (such as LAT-to-TCP or TCP-to-LAT) and converting one virtual terminal protocol into another protocol.

For more information about these services, refer to the “Access Services Overview” chapter in this publication. Figure 19 shows the connection services available on your router.

**Figure 19 Connection Services**



## Connection Task List

To make, monitor, or manage connections, complete the relevant tasks in the following sections:

- Make Remote Node Connections—Describes PPP, SLIP, XRemote, and ARA.
- Make Terminal Service Connections—Describes Telnet and UNIX rlogin, IPX dial-out, LPD printer, LAT, TN3270, and mobile remote node connections.
- Make Protocol Translation Connections—Describes one-step and two-step protocol translation methods, and X.3 PAD connections.

- Monitor Connections—Describes various types of information used to monitor connections.
- Manage Connections—Describes management tasks with connections, including escaping to the EXEC prompt, switching sessions, exiting sessions, setting X.3 PAD parameters, disconnecting a line, and so on.
- Change Terminal Session Parameters—Describes how to change terminal and line settings temporarily for each connection session. The local settings temporarily override those made by the system administrator, remaining in effect only until the user exits the system.

For examples of these connections and tasks, refer to the last section in this chapter, “Connection Examples.”

## Make Remote Node Connections

This section describes how to connect devices across telephone lines by using the Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or XRemote (the NCD X Windows terminal protocol). The section also describes how Macintosh users can connect to an AppleTalk network through AppleTalk Remote Access (ARA).

This section contains the following sections:

- PPP Connections
- SLIP Connections
- XRemote Connections
- ARA Connections

### PPP Connections

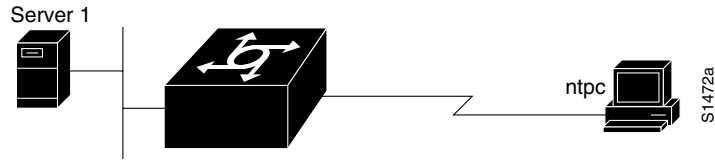
When you connect from a remote node computer, through an asynchronous port on an access server to the EXEC facility, and you want to connect from the access server to a device on the network, perform the following task in EXEC mode:

Task	Command
Create a PPP connection.	<b>ppp /default</b>   <i>{remote-ip-address   remote-name}</i> <i>[@tacacs-server] [/routing]</i>

If you specify an address for the TACACS server using **/default** or *tacacs-server*, the address must be the first parameter in the command after you type **ppp**. If you do not specify an address or enter **/default**, you are prompted for an IP address or host name. You can enter **/default** at this point.

For example, in Figure 20, if you are working at home on the device *ntpc* and want to connect to Server 1 using PPP, you could dial into the access server. When you connect to the EXEC prompt on the access server, type the **ppp** command to connect with the device.

Figure 20 Using the PPP EXEC Command



To terminate a session, disconnect from the device on the network using the command specific to that device. Then, exit from the EXEC by using the **exit** command.

For an example of making a PPP connection, see the “PPP Connection Example” section at the end of this chapter.

## SLIP Connections

To make a serial connection to a remote host by using SLIP, perform the following task in EXEC mode:

Task	Command
Create a SLIP connection.	<b>slip /default</b>   { <i>remote-ip-address</i>   <i>remote-name</i> } [ <i>@tacacs-server</i> ] [ <i>/routing</i> ] [ <i>/compressed</i> ]

Your system administrator can configure SLIP to expect a specific address or to provide one for you. It is also possible to set up SLIP in a mode that compresses packets for more efficient use of bandwidth on the line.

If you specify an address for the TACACS server using **/default** or *tacacs-server*, the address must be the first parameter in the command after you type **slip**. If you do not specify an address or enter **/default**, you are prompted for an IP address or host name. You can enter **/default** at this point.

If you do not use the *tacacs-server* argument to specify a TACACS server for SLIP address authentication, the TACACS server specified at login (if any) is used for the SLIP address query.

To optimize bandwidth on a line, SLIP enables compression of the SLIP packets using Van Jacobson TCP header compression as defined in RFC 1144.

To terminate a session, disconnect from the device on the network using the command specific to that device. Then, exit from EXEC mode by using the **exit** command.

For an example of making a SLIP connection, see the “SLIP Connection Examples” section at the end of this chapter.

## XRemote Connections

You use the XRemote protocol with an X display station and a modem to connect to remote hosts via TCP/IP and LAT. This section outlines the steps for starting XRemote in several typical environments and for exiting XRemote sessions. It contains the following sections:

- Connect through Automatic Session Startup with an XDMCP Server.
- Connect through Automatic Session Startup with a DECwindows Login via LAT.
- Connect through Manual XRemote Session Startup.

- Establish XRemote Sessions between Servers.
- Exit XRemote Sessions.

When possible, use the automated processes. Make sure that your system administrator has already configured a path for loading fonts.

You can run the XRemote protocols between two servers. This is useful if you use an X display server that does not support XRemote, or if an X display station is connected to a LAN and you want to use the LAN rather than a dial-in link to connect to a server. (Note that XRemote is faster when the X display station connects to a server over a dial-in link.) See the section, “Establish XRemote Sessions between Servers.”

For an example of making an XRemote connection, see the “XRemote Examples” section at the end of this chapter.

### Connect through Automatic Session Startup with an XDMCP Server

If your host computer supports a server for XDMCP (such as the `xdm` program included in X11R4 or later), you can use automatic session startup to make an XRemote session connection. To do so, perform the following task in EXEC mode:

Task	Command
Create a connection with XRemote and an XDMCP server.	<code>xremote xdm [hostname]</code>

This command sends an XDMCP session startup request to the host computer. If you do not specify a host name, a broadcast message is sent to all hosts. The first host to respond by starting up a session is used.

The server and X terminal stay in XRemote mode until either the display manager terminates the session, or a reset request is received from the X terminal.

### Connect through Automatic Session Startup with a DECwindows Login via LAT

If your host computer supports DECwindows login sessions, you can use automatic session startup to make an XRemote session connection. If the system administrator at the remote host configures support for DECwindows over LAT, perform the following task in EXEC mode to initiate the connection:

Task	Command
Create a connection with XRemote and DECwindows over LAT.	<code>xremote lat service</code>

After you issue this command, expect the following to occur:

- The XRemote font server loads several initial fonts for the DECwindows login display.
- The terminal displays the Digital logo and DECwindows login box.

Log on to the system. Upon completion of login, more fonts are loaded, and the remote session begins.

**Note** Because of heavy font usage, DECwindows applications can take longer than expected to start when you use XRemote. After the application starts, performance and access times should be normal.

### Connect through Manual XRemote Session Startup

If you do not use a host computer that supports XDMCP or LAT, you must use manual session startup. To use manual session startup, perform the following tasks in EXEC mode:

- Enable XRemote manually on the server port.
- Connect to the remote host computer.
- Set the location of the X display.
- Start client applications.
- Return to the EXEC prompt.
- Enable XRemote manually again on the server port.

The following sections describe these tasks.

#### Enable XRemote Manually

To prepare the XRemote server for manual startup, perform the following task in EXEC mode:

Task	Command
Prepare the XRemote server for manual startup.	<b>xremote</b>

After you issue this command, instructions prompt you through the process of manually enabling XRemote.

**Note** In manual operation, the server and X terminal remain in XRemote mode until all clients disconnect or the server receives a reset request from the X terminal. A session might terminate during startup because you invoked transient X clients that set some parameters and then disconnected (such as **xset** or **xmodmap** parameters). There must always be one session open or the connection is reset.

#### Connect to the Remote Host Computer

To connect to a host, perform one of the following tasks in EXEC mode:

Task	Command
Prepare the server for XRemote manual startup.	<b>telnet</b> or <b>lat</b> or <b>rlogin</b>

After entering the command, you can log on as usual.

## Set the Location of the X Display

At this point, you are logged in to the remote host computer.

---

**Note** If you are using a version of Telnet on the remote host that supports the “X Display Location” option (RFC 1096), skip this step and go on to the “Start Client Applications” section.

---

Inform the host computer of your X display location that the server provided when you enabled XRemote manually.

For most versions of the UNIX operating system, the X display location is set by using the **setenv** command to set the Display environment variable. Refer to your UNIX system’s online X(1) manual page for more information.

On VAX/VMS systems, use the **SET DISPLAY** command to set the X display location. For more information, refer to the *VMS DCL Dictionary*.

---

**Note** To set the location of the X display for VAX/VMS client systems, you must install either the TCP/IP transport from Digital or a third-party TCP/IP transport. Contact your VAX/VMS system administrator for the appropriate TCP/IP transport name.

---

## Start Client Applications

Now you can start your client applications for your host operating system, as specified in the documentation for the client applications.

The server accepts the X connection attempt from the client application and places the client in a dormant state.

## Return to the EXEC Prompt

If it is possible to log off the host computer and keep your X clients running in the background, you can do so now. This conserves resources on both the host and the server that would otherwise be inaccessible until you exited from the XRemote state.

If you cannot log off the host computer and keep your clients running, escape back to the access server’s EXEC prompt using the escape sequence (**Ctrl-Shift-6** then **x [Ctrl^x]** by default).

## Re-enable XRemote Manually

To begin a manual remote session again, refer to the “Enable XRemote Manually” section earlier in this chapter. If the X clients connected successfully, the session is put into XRemote mode, and the clients complete their startup.

If no clients are found, you see the following message:

```
No X clients waiting - check that your display is darkstar:2018
```

Check your hosts to determine whether an error has occurred when the session started. The most likely causes are that there is an improperly specified display location, or the host computer did not recognize the name of your server.

### Establish XRemote Sessions between Servers

If you are on an X display server that does not support XRemote, you can still run the XRemote protocols. An X display server (such as a PCX, MacX, or UNIX workstation) connected to an Ethernet network can dial out through an access server on a conventional modem to access an X client program on a host residing on another network. The access server provides the server-side helper process.

To run XRemote, connect to one of the XRemote ports.

---

**Note** The NCD helper process does not support X display devices that use a maximum request and response size larger than 64 Kb.

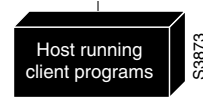
---

Find out from your administrator whether the connection from your X display server is configured as an individual line or a rotary connection.

- To connect to an individual line, use Telnet to connect from the X display server to port 9000 plus the decimal value of the line number.
- To make a rotary connection, use Telnet to connect from the X display server to port 10000 plus the decimal value of the line number.

For information about how to configure individual lines and rotary connections, refer to the “Configuring Terminal Lines and Modem Support” chapter earlier in this publication, and refer to the *Access Services Command Reference*.

Figure 21 illustrates a configuration in which a display server is not running XRemote. In this configuration, the server-side XRemote helper is running on Access Server 1, and the client-side XRemote helper is running on Access Server 2.

**Figure 21 XRemote Session between Servers**

## Exit XRemote Sessions

When you exit XRemote, you must quit all active X connections, usually with a command supported by your X client system. Usually, when you quit the last connection (all client processes are stopped), XRemote closes and you return to the EXEC prompt. Check your X client system documentation for specific information about exiting an XRemote session.

## ARA Connections

If you are a Macintosh user, you can use AppleTalk Remote Access (ARA) to connect to an AppleTalk network through a Cisco access server. The Cisco IOS Release 10.2 and later software supports ARA 2.0 and ARA 1.0 so that you can remotely dial in through asynchronous network devices using ARA to access AppleTalk services (such as file sharing and printing) elsewhere on the network. For example, you can dial in from an X.25 network and connect to an AppleTalk network through a router. To enable ARA and dial-in access, configure a virtual terminal (VTY) line on the router. You can also configure ARA on TTY lines.

Because there are no user commands for connecting to the network from your Macintosh client, the process is not described in this publication. To start a connection in most ARA client packages, you click the Connect button from within the client software.

## Make Terminal Service Connections

This section describes how to connect a client terminal or microcomputer running terminal-emulation software such as Telnet, rlogin, TN3270, and Local Area Transport (LAT) through Cisco routers to a host across a LAN or WAN. Specifically, this section contains the following sections:

- Telnet and UNIX rlogin Connections
- IPX Dial-out Connections
- LPD Printer Connections
- LAT Connections
- TN3270 Connections
- Mobile Remote Node Connections

### Telnet and UNIX rlogin Connections

Telnet and rlogin are protocols that enable TCP/IP connections to a host.

Telnet, a virtual terminal protocol that is part of the TCP/IP protocol suite, is the more widely used protocol.

The rlogin protocol is a remote login service developed for the BSD UNIX system. It provides better control and output suppression than Telnet, but can only be used when the host (typically, a UNIX system) supports rlogin. The Cisco IOS implementation of rlogin does not subscribe to the rlogin “trusted host” model. That is, a user cannot automatically log on to a UNIX system from the router, but must provide a user ID and a password for each connection.

The Cisco IOS implementation of Telnet and rlogin provides the connection capabilities described in the following sections:

- Make Telnet Connections.
- Execute Special Telnet Escape Sequences.
- Make rlogin Connections.
- Switch between Sessions.
- Exit Telnet and rlogin Sessions.

### Make Telnet Connections

To log on to a host that supports Telnet, perform the following task in EXEC mode:

Task	Command
Log on to a host that supports Telnet.	<b>connect</b> <i>host</i> [ <i>port</i> ] [ <i>keyword</i> ] or <b>telnet</b> <i>host</i> [ <i>port</i> ] [ <i>keyword</i> ]

With the Cisco IOS implementation of TCP/IP, you are not required to enter the **connect** or **telnet** commands to establish a Telnet connection. You can just enter the learned host name—as long as the host name is different from a command word for the router. Telnet must be the default (you can make it the default with the **transport preferred** command. See the “Terminal Lines and Modem Support Commands” chapter in the *Access Services Command Reference*).

To display a list of the available hosts, perform the following task in EXEC mode:

Task	Command
Display a list of available hosts.	<b>show hosts</b>

To display the status of all TCP connections, perform the following task in EXEC mode:

Task	Command
Display the status of all TCP connections.	<b>show tcp</b>

The Cisco IOS software assigns a logical name to each connection, and several commands use these names to identify connections. The logical name is the same as the host name, unless that name is already in use, or you change the connection name with the **name-connection** EXEC command. If the name is already in use, the Cisco IOS software assigns a null name to the connection.

For an example of making a Telnet connection, see the “Telnet Connection Examples” section at the end of this chapter.

## Execute Special Telnet Escape Sequences

The Telnet software supports special Telnet commands in the form of Telnet sequences that map generic terminal control functions to operating system-specific functions.

To issue a special Telnet command, enter the escape sequence followed by the command character. The default escape sequence is Ctrl^ (press and hold the Control and Shift keys while pressing the 6 key). You can enter the command character as you hold down Ctrl or with Ctrl released; you can enter the command characters as either uppercase or lowercase letters.

Table 5 lists the special Telnet escape sequences.

**Table 5 Special Telnet Escape Sequences**

Task	Escape Sequence <sup>1</sup>
Break	<b>Ctrl^ b</b>
Interrupt Process (IP)	<b>Ctrl^ c</b>
Erase Character (EC)	<b>Ctrl^ h</b>
Abort Output (AO)	<b>Ctrl^ o</b>
Are You There? (AYT)	<b>Ctrl^ t</b>
Erase Line (EL)	<b>Ctrl^ u</b>

1. Pressing Ctrl displays a caret (^) character.

At any time during an active Telnet session, you can list the Telnet commands by pressing the escape sequence keys (by default **Ctrl-Shift-6**) followed by a question mark at the system prompt:

**Ctrl-^ ?**

A sample of this list follows.

```
Router> ^^?
```

---

**Note** In screen output examples that show two caret (^) symbols together, the first caret represents the Control key and the second caret represents the keystroke sequence Shift-6. The double caret combination (^^) means hold down the Control key while you press the Shift and the 6 key.

---

```
[Special telnet escape help]
^^B  sends telnet BREAK
^^C  sends telnet IP
^^H  sends telnet EC
^^O  sends telnet AO
^^T  sends telnet AYT
^^U  sends telnet EL
```

### Make rlogin Connections

To log on to a UNIX host by using rlogin, perform the following task in EXEC mode:

Task	Command
Log on to a host that supports rlogin.	<b>rlogin</b> <i>host</i> [ <i>debug</i> ] [ <i>/user username</i> ]

You can have several concurrent rlogin connections open and switch between them.

To open a new connection, exit the current connection by entering the escape sequence (**Ctrl-Shift-6** then **x** [**Ctrl^x**] by default) to return to the system command prompt, then open a new connection.

For an example of making a rlogin connection, see the “rlogin Example” section at the end of this chapter.

### Switch between Sessions

You can have several concurrent sessions open and switch back and forth between them. The number of sessions that can be open is defined by the **session-limit** command, which is described in the “Configuring Terminal Lines and Modem Support” chapter in this publication, and in the “Terminal Lines and Modem Support Commands” chapter of the *Access Services Command Reference*.

To switch between sessions by escaping one session and resuming a previously opened session, perform the following tasks. The following commands work for both Telnet and rlogin sessions:

Task	Command
<b>Step 1</b> Escape out of the current session and return to the EXEC prompt.	<b>Ctrl-Shift-6</b> then <b>x</b> ( <b>Ctrl^x</b> ) by default
<b>Step 2</b> List the open sessions. All open sessions associated with the current terminal line are displayed.	<b>where</b>
<b>Step 3</b> Make the connection.	<b>resume</b> [ <i>connection</i> ] [ <i>keyword</i> ]

The **Ctrl^x**, **where**, and **resume** commands are available with all supported connection protocols.

For an example of switching between sessions, see the “Switch between Telnet and rlogin Sessions Examples” section at the end of this chapter.

## Exit Telnet and rlogin Sessions

To exit a Telnet or rlogin session, type the escape sequence (**Ctrl-Shift-6** then **x** [**Ctrl^x**] by default) and enter the **disconnect** command at the EXEC prompt. You can also log off the remote system to exit a session.

You also can issue any of the following commands to at the EXEC prompt to terminate an active Telnet or rlogin session:

```
exit
logout
```

## IPX Dial-out Connections

If you have configured NetWare asynchronous services interface (NASI) on your router, you can use IPX client applications to make IPX dial-out connections to a shared pool of asynchronous devices. For example, a NASI client on the LAN can connect to a serial (synchronous or asynchronous) port on the router, which provides access to remote modems, printers, and networks. The command the user issues depends on the application being used to connect to the NASI server.

NASI relies on Sequenced Packet Exchange (SPX).

To configure your router as a NASI server, refer to the section “Enable NASI Clients to Access Network Resources” in the “Configuring Terminal Lines and Modem Support” chapter earlier in this publication.

## LPD Printer Connections

The Berkeley UNIX Line Printer Daemon (LPD) protocol is used to send print jobs between UNIX systems. The Cisco IOS software supports a subset of the LPD protocol that provides the following features:

- Improved status information
- Cancellation of printer jobs
- Confirmation of successful printing and automatic retry for common failures
- Use of standard UNIX software

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, perform the following task in EXEC mode:

Task	Command
List currently defined printers and their usage statistics.	<b>show printer</b>

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 */etc/printcap* on your UNIX system to include the definition of the remote printer in the Cisco IOS software. For more information on setting up your printer, refer to the “Configuring Terminal Lines and Modem Support” chapter earlier in this publication.

## LAT Connections

The Digital Equipment Corporation (Digital) LAT protocol is most often used to connect routers to Digital hosts. LAT is a Digital-proprietary protocol, and the Cisco IOS software uses LAT technology licensed from Digital to perform the LAT connection tasks described in the following sections:

- Make a LAT Connection.
- Define a Group Code List for Outgoing LAT Connections.
- Switch between LAT Sessions.
- Use Digital Commands on the Server.
- Exit a LAT Session.

### Make a LAT Connection

To connect to a LAT host, perform the following task in EXEC mode:

Task	Command
Connect to a LAT host.	<b>lat</b> <i>name</i> [ <b>node</b> <i>nodename</i>   <b>port</b> <i>portname</i>   <b>/debug</b> ]

You can quit the connection by pressing **Ctrl-C** or complete the connection by entering the password for a given service.

You can also set your preferred connection protocol to any available connection protocol supported in the Cisco IOS software. Your preferred connection protocol is also referred to in the Cisco IOS software as a “preferred transport type.” If your preferred connection protocol is set to **lat**, you can use the **connect** command in place of the **lat** command. To configure a preferred connection protocol, use the **transport preferred** command as described in the “Configuring Terminal Lines and Modem Support” chapter earlier in this publication. When your preferred connection protocol is set to **none** or to another protocol, you must use the **lat** command to connect to a LAT host.

For an example of making a LAT connection, see the “LAT Connection Examples” section at the end of this chapter.

### Define a Group Code List for Outgoing LAT Connections

To specify a temporary list of services to which you or another user can connect, you must define the group code lists used for connections from specific lines. You limit the connection choices for an individual line by defining the group code lists for an outgoing connection. To define a group code list, perform the following task in EXEC mode:

Task	Command
Define a temporary list of services to which you or another user can connect by defining the group code lists used for connections from specific lines.	<b>terminal lat out-group</b> { <i>groupname</i>   <i>number</i>   <i>range</i> }

When a user initiates a connection with a LAT host, the user's line must share a common group number with the remote LAT host before a connection can be made. The group code range *must be* a subset of the line's configured group code range.

For an example of defining a group code list, see the "Define a Group Code List for Outgoing LAT Connections Example" section at the end of this chapter.

## Switch between LAT Sessions

You can have several concurrent LAT sessions open and switch between them.

To open a subsequent session, first enter the escape sequence (**Ctrl-Shift-6** then **x** [**Ctrl^x**] by default) to suspend the current session. Then open a new session.

To list the available LAT services, perform the following task in EXEC mode:

Task	Command
List available LAT services.	<b>show lat services</b>

For sample output of this **show** command, refer to the "Monitor Connections" section later in this chapter.

## Use Digital Commands on the Server

To view the subset of Digital commands that the Cisco IOS software supports, perform the following task in EXEC mode:

Task	Command
List EXEC commands.	<b>help</b>

## Exit a LAT Session

To exit a session, simply log off the remote system. Then, terminate an active LAT session by entering the **exit** or **logout** command.

## TN3270 Connections

You use TN3270 terminal emulation to connect to an IBM 3278-type host. Your system administrator must configure a default terminal emulation file that permits the terminal to communicate with the host. How to specify alternate terminal emulations is described in the "Configuring TN3270" chapter in this publication and in the *Access Services Command Reference*. Your administrator can also specify custom terminal emulations.

Unlike Telnet and LAT connections, you *must* enter the **tn3270** command to make a connection to an IBM 3278 host. To begin a TN3270 session, perform the following task in EXEC mode:

Task	Command
Begin a TN3270 connection.	<b>tn3270 host</b>

To terminate an active TN3270 session, enter the escape sequence (**Ctrl-Shift-6** then **x** [**Ctrl^x**] by default) and enter the **disconnect** command at the EXEC prompt. You can also log off the remote system by issuing the command specific to that system (such as **exit**, **logout**, **quit**, **close**, or **disconnect**).

For an example of setting TN3270 connections, see the “TN3270 Connection Example” section at the end of this chapter.

## Mobile Remote Node Connections

If you are a mobile user, dialing in to your “home” router from a remote site is often impractical. The asynchronous mobility feature enables you to dial in to different routers elsewhere on the internetwork while experiencing the same server environment that you would if you were connecting directly to your home router.

This asynchronous host mobility is accomplished by *packet tunneling*, a technique by which raw data from the dial-in user is encapsulated and transported directly to the host site where your home router performs the actual protocol processing.

Asynchronous mobility requires setting up a network layer connection between two Cisco routers—the home and remote router. To do so, perform the following task in user EXEC mode:

Task	Command
Set up a network layer connection to a router by specifying its Internet name or address.	<code>tunnel host</code>

From a router other than a Cisco router, you must use Telnet.

After a connection is established, you receive an authentication dialog or prompt from your home router, and can proceed as if you are connected directly to that router. When communications are complete, the network connection can be closed and terminated from either end of the connection.

The home router must be configured correctly to accept the tunnel connection.

For an example of setting mobile remote node connections, see the “Mobile Remote Node Example” section at the end of this chapter.

## Make Protocol Translation Connections

This section describes the methods you can use to connect a host running one protocol (such as Telnet with TCP/IP) to a host running another protocol (such as LAT). This process is called *protocol translation* and allows devices running different protocols (such as X.25 and TCP/IP) to communicate. Protocol translation does not permit translation between other services such as file transfer protocols.

You can make a protocol translation connection using the protocols listed in the section “Protocol Translation Methods.” The commands you use to make these connections and exit from them are listed in the “Connection Commands” chapter of the *Access Services Command Reference*.

This section describes the additional tasks required to perform protocol translation from one host to another host or to a router. Specifically, it contains the following sections:

- Protocol Translation Methods
- X.3 PAD Connections

## Protocol Translation Methods

The Cisco IOS software supports virtual terminal connections in both directions between the protocols in the following list. You can configure the router to translate automatically between them. This is called *one-step translation*.

- X.25 to Local Area Transport (LAT)
- X.25 to Telnet sessions using the Transmission Control Protocol (TCP)
- LAT to TCP/Telnet
- LAT, X.25, and TCP (Telnet) to SLIP and PPP

The Cisco IOS supports limited connections in both directions between the following protocols. Connecting between these protocols requires that you first connect to a router, then to the host to which you want to connect. This is called *two-step translation*.

- XRemote to SLIP/PPP and X.25 PAD environments
- LAT, X.25, SLIP/PPP, and TCP (Telnet) to TN3270

The following sections describe one-step and two-step protocol translation.

For an example of protocol translation connections, see the “Protocol Translation Session Examples” section at the end of this chapter.

### One-Step Protocol Translation

In general, you use one-step protocol translation when network users repeatedly log on to the same remote network hosts through a router. This connection is more efficient and enables the Cisco IOS software to monitor the protocols in use because the router acts as a network connection rather than as a terminal.

One-step protocol translation is transparent. When connecting to the remote network host, the user enters the **translate** command to the remote network host, but need not specify protocol translation. The network administrator creates a configuration that defines a connection and the protocols to be translated. The user performs one step to connect with the host.

When you make a one-step connection to the router, the Cisco IOS software determines which host the connection is for and which protocol that host is using. It then establishes a new network connection using the protocol required by that host.

To support connections in each direction, the network administrator must include **translate** command statements in the configuration file. Refer to this publication and the *Access Services Command Reference* for information about configuring the Cisco IOS software for one-step connections and for information about the **translate** command.

A disadvantage of one-step protocol translation is that the initiating computer or user does not know that two networking protocols are being used. This means that parameters of the foreign network protocols cannot be changed after connections are established. The exception to this limitation is the set of parameters common to both networking protocols. Any common parameter can be changed from the first host to the final destination.

### Two-Step Protocol Translation

In general, you use two-step protocol translation when you want to use protocol translation for one-time connections. The network administrator must have first configured the Cisco IOS software for the connection protocols you are using.

With the two-step connection process, you can modify the parameters of either network connection, even while a session is in process. This process is similar to connecting a group of terminal lines from a packet assembler/disassembler (PAD) to a group of terminal lines from a TCP router. The difference is that when two devices are connected via asynchronous serial lines, you do not encounter wiring complexity, unreliability, management problems, and performance bottlenecks.

Also, two-step protocol translation allows another level of security over the one-step translation method when TACACS password protection is enabled. These security features are described in the *Security Configuration Guide*, which is a publication in the Cisco IOS configuration guides and command references.

To connect to the remote network host running a foreign protocol, perform the following steps:

- Step 1** Make a network connection to a router by using the EXEC command for the protocol running at your local terminal.  
  
These commands are listed in the “Connection Commands” chapter of the *Access Services Command Reference*. X.3 PAD connections are described later in this chapter.
- Step 2** When the system prompt appears, connect to the remote host using the EXEC command for the protocol running on the remote host (such as TN3270 or XRemote).

### X.3 PAD Connections

A packet assembler/disassembler (PAD) is a device that receives a character stream from one or more terminals, assembles the character stream into packets, and sends the data packets out to a host. A PAD can also do the reverse. It can take data packets from a network host and translate them into a character stream that can be understood by the terminals. A PAD is defined by CCITT Recommendations X.3, X.28, and X.29. The following sections describe X.3 PAD connection tasks:

- Make a PAD Connection.
- Switch between Connections.
- Exit a PAD Session.

For an example of X.3 PAD connections, see the “X.3 PAD Session Example” section at the end of this chapter.

#### Make a PAD Connection

To log on to a PAD, perform the following task in EXEC mode:

Task	Command
Log on to a PAD.	<code>pad {X.121-address   hostname} [/cud text] [/debug] [/profile name] [/reverse]</code>

To display information about packet transmission and X.3 PAD parameter settings, perform the following task in EXEC mode:

Task	Command
Display information about packet transmission and X.3 PAD parameter settings.	<code>show x25 pad</code>

You can exit a connection and return to the user EXEC prompt at any point.

To open a new connection, first exit the current connection by typing the escape sequence (**Ctrl-Shift-6** then **x** [**Ctrl^x**] by default) to return to the EXEC prompt, then open the new connection.

The **show x25 pad** command is described in the “Monitor Connections” section later in this chapter.

## Switch between Connections

You can have several concurrent sessions open and switch between them. The number of sessions that can be open is defined by the **session-limit** command, which is described in the “Configuring Terminal Lines and Modem Support” chapter of this publication and the “Terminal Lines and Modem Support Commands” chapter of the *Access Services Command Reference*.

To switch between sessions by escaping one session and resuming a previously opened session, perform the following tasks:

Task	Command
<b>Step 1</b> Escape the current connection and return to the EXEC prompt.	<b>Ctrl-Shift-6</b> then <b>x</b> ( <b>Ctrl^x</b> ) by default
<b>Step 2</b> List the open sessions. All open sessions associated with the current terminal line are displayed.	<b>where</b>
<b>Step 3</b> Make the connection.	<b>resume</b> [ <i>connection</i> ] [ <i>keyword</i> ]

The **Ctrl^x**, **where**, and **resume** commands are available with all supported connection protocols.

## Exit a PAD Session

To exit a PAD session, enter the escape sequence (**Ctrl-Shift-6** then **x** [**Ctrl^x**] by default) and enter the **disconnect** command at the EXEC prompt. You can also log off the remote system by issuing the command specific to that system (such as **exit**, **logout**, **quit**, **close**, or **disconnect**).

## Monitor Connections

This section describes the procedures that you use to monitor network devices and activities, including viewing the status of all sessions or active ports on the routers.

The following sections describe the procedures used to monitor network devices and activities for the following connection protocols:

- Monitoring Connections Generic to All Connections
- Monitoring LAT Connections
- Monitoring TCP/IP Connections
- Monitoring XRemote Connections
- Monitoring X.25 PAD Connections

**Note** The various **show** commands described in these sections are only a subset of the available monitoring commands for each protocol. For information about additional monitoring commands for a given protocol, refer to the command reference chapter in the *Access Services Command Reference* for the protocol.

Enter all commands for monitoring connections at the user level EXEC prompt.

## Monitoring Connections Generic to All Connections

This section describes commands that are generic to all connection protocols. It has the following sections:

- View Open Connections.
- View Network Hosts.
- View Active Router Lines.
- View Queued Host-initiated Connections.
- View Terminal Lines.

### View Open Connections

To display information about open connections, perform the following task in user EXEC mode:

Task	Command
Display information about open connections.	<b>show sessions</b>
	or
	<b>where</b>

These commands display the host name, address, number of unread bytes, idle time, and connection name.

### View Network Hosts

To display the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of host names and addresses on the network to which you have connected, perform the following task in user EXEC mode:

Task	Command
Display host information.	<b>show hosts</b>

## View Active Router Lines

To display information about the active lines on the router, perform the following task in user EXEC mode:

Task	Command
Display router active-line information.	<b>show users [all]</b> or <b>systat [all]</b>

These commands display the same information, including the line number, connection name, idle time, and terminal location.

## View Queued Host-initiated Connections

To display the list of queued host-initiated connections to a router, perform the following task in user EXEC mode:

Task	Command
Display the list of queued host-initiated connections to a router.	<b>show entry</b>

You can use this command to identify which LAT hosts have queue entries for printers.

## View Terminal Lines

To display local terminal settings, perform the following task in user EXEC mode:

Task	Command
Display local terminal settings.	<b>show terminal</b>

This command displays information about the current terminal line (such as the line number, line status, modem state, and special characters set). This information is useful for changing lines to match expected settings using local terminal parameter-setting tasks described in the “Configuring Terminal Lines and Modem Support” chapter earlier in this publication.

The display includes a comprehensive report on the terminal settings in effect, including the preferred connection protocol.

## Monitoring LAT Connections

The following sections describe the commands used to monitor LAT connections:

- Show LAT Learned Services.
- Show Active LAT Sessions.

### Show LAT Learned Services

To display information on all LAT-learned services, perform the following task in user EXEC mode:

Task	Command
Display LAT-learned services.	<b>show lat services</b>

To display information on LAT-learned services in a DECserver format, perform the following task in user EXEC mode:

Task	Command
Display LAT-learned services.	<b>show service</b> [ <i>service-name</i> ]

The **show service** command without the service name argument displays a list of known LAT-learned services. When entered with the *service-name* argument, it displays a more detailed status of the named service. If there is no LAT-learned service by the specified name, the Cisco IOS software looks for an IP host with that name.

### Show Active LAT Sessions

To display active LAT sessions, perform the following task in user EXEC mode:

Task	Command
Display active LAT sessions.	<b>show lat sessions</b> [ <i>line-number</i> ]

### Monitoring TCP/IP Connections

To display the status of a TCP connection, perform the following task in user EXEC mode:

Task	Command
Display the status of a TCP connection.	<b>show tcp</b> [ <i>line-number</i> ]

To view a summary of the TCP connection end points in the system, perform the following task in user EXEC mode:

Task	Command
Display a summary of the TCP connection end points in the system.	<b>show tcp brief</b> [ <i>all</i> ]

### Monitoring XRemote Connections

To list XRemote connections and monitor XRemote traffic through the router, perform the following task in user EXEC mode:

Task	Command
List XRemote connections and monitor XRemote traffic through the networking hardware.	<b>show xremote</b>

This command provides XRemote parameters applied to the entire Cisco IOS configuration, and statistics that are pulled for all active XRemote connections.

To list XRemote connections and monitor XRemote traffic for specific lines on an XRemote server, perform the following task in user EXEC mode:

Task	Command
List XRemote connections and monitor XRemote traffic for specific lines on a server.	<b>show xremote line</b> <i>number</i>

## Monitoring X.25 PAD Connections

To display information about current open connections, perform the following task in user EXEC mode:

Task	Command
Display information about X.25 PAD connections that are currently open.	<b>show x25 pad</b>

The information includes packet transmissions, X.3 parameter settings, and the current status of virtual circuits.

## Manage Connections

This section describes session management activities. The following sections describe connection management activities that apply to all supported connection protocols:

- Escape to the EXEC Prompt
- Switch to Another Connection
- Assign a Logical Name to a Connection
- Change a Login Name
- Lock Access to a Terminal
- Specify a TACACS Host
- Set X.3 PAD Parameters
- Send Messages to Other Terminals
- Exit a Session Started from a Router
- Log Out of a Router
- Disconnect a Line
- Clear TCP/IP Connections

### Escape to the EXEC Prompt

After you have started a connection, you can escape out of the current session and return to the EXEC prompt by using the escape sequence command (**Ctrl-Shift-6** then **x** [**Ctrl^x**] by default). You can type the command character as you hold down Ctrl or with Ctrl 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 and the second caret represents the keystroke sequence Shift-6. The double caret combination (^) means hold down the Control key while you press the Shift and the 6 key.

---

By default, the escape sequence is **Ctrl^x**. If you press the escape key (**Escape-Char**), you change the **Shift-Ctrl-6** sequence to whatever you want. For example, if you press **Escape-Char Break**, the **Break** key becomes the new escape character to suspend a session and access the EXEC prompt.

## Switch to Another Connection

You can have several concurrent sessions open and switch back and forth between them.

The number of sessions that can be open is defined by the **session-limit** command, which is described in the “Configuring Terminal Lines and Modem Support” chapter earlier in this publication and the “Terminal Line and Modem Support Commands” chapter of the *Access Services Command Reference*.

To switch between sessions by escaping one session and resuming a previously opened session, perform the following tasks:

Task	Command
<b>Step 1</b> Escape the current connection and return to the EXEC prompt.	<b>Ctrl-Shift-6</b> then <b>x</b> ( <b>Ctrl^x</b> ) by default
<b>Step 2</b> List the open sessions. All open sessions associated with the current terminal line are displayed.	<b>where</b>
<b>Step 3</b> Make the connection.	<b>resume</b> [ <i>connection</i> ] [ <i>keyword</i> ]

The **Ctrl^x**, **where**, and **resume** commands are available with all supported connection protocols.

You could also make a new connection while you are at the EXEC prompt.

## Assign a Logical Name to a Connection

To assign a logical name to a connection, perform the following task in EXEC mode:

Task	Command
Assign a logical name to a connection.	<b>name-connection</b>

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** command displays a list of the assigned logical connection names.

## Change a Login Name

You can change a login username if you must match outgoing access list requirements or other login prompt requirements. To change a login username, perform the following task in user EXEC mode:

Task	Command
Change a login username.	<b>login</b> <sup>1</sup>

1. This command is documented in the *Security Command Reference*.

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 that the **login** command attempt used.

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.

When you access a system with TACACS security, you can enter your login name or specify a TACACS server by using the following argument when the “Username:” prompt appears:

*user @tacacs-server*

The router must be one of the routers defined in a router configuration. For more information, refer to the “Specify a TACACS Host” section later in this chapter, or refer to the **tacacs-server host** command in the “Network Access Security Commands” chapter of the *Security Command Reference*.

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 denies access or function, according to the action specified by the **tacacs-server last-resort** 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 SLIP address queries.

For an example of changing a login name, see the “Change a Login Name Example” section at the end of this chapter.

## Lock Access to a Terminal

You can prevent access to your terminal session while keeping your connection open by setting up a temporary password. To lock access to the terminal, perform the following tasks in EXEC mode:

Task	Command
<b>Step 1</b> Issue the <b>lock</b> command. The system prompts you for a password.	<b>lock</b>
<b>Step 2</b> Enter a password, which can be any arbitrary string. The screen clears and displays the message “Locked.”	<i>password</i>
<b>Step 3</b> To regain access to your sessions, re-enter the password.	<i>password</i>

The Cisco IOS software honors session timeouts on a locked line. You must clear the line to remove this feature. The system administrator must set up the line to allow use of the temporary locking feature.

## Specify a TACACS Host

You can specify a TACACS host when you dial in or use the **login** command. Only the specified host is accessed for user authentication information.

To specify the name of a TACACS host at login, perform the following tasks in EXEC mode:

Task	Command
Specify the name of a TACACS host at login.	<code>user@hostname</code>

For an example of specifying a TACACS host, see the “Specify a TACACS Host Example” section at the end of this chapter.

## Set X.3 PAD Parameters

To set X.3 PAD parameters, perform one of the following tasks in EXEC mode:

Task	Command
Set X.3 PAD parameters.	<code>resume [connection] [/set parameter:value]</code> or <code>x3 parameter:value</code>

The parameters are numbered from 1 through 18. See the “X.3 PAD Parameters” appendix of the *Access Services Command Reference* for more information.

For an example of setting X.3 PAD parameters, see the “Set X.3 PAD Parameters Example” section at the end of this chapter.

## Send Messages to Other Terminals

You can send messages to one or all terminal lines. A common reason for doing this is to inform users of an impending shutdown. To send a message to other terminals, perform the following task in EXEC mode:

Task	Command
Send a message to other terminals.	<code>send {line-number   *}</code>

The system prompts for the message, which can be up to 500 characters long. Enter **Ctrl-Z** to end the message. Enter **Ctrl-C** to abort the command.

## Exit a Session Started from a Router

The protocol used to initiate a session determines how you exit that session.

To exit XRemote, you must quit all active X connections, usually with a command supported by your X client system. Usually, when you quit the last connection (all client processes are stopped), XRemote closes and you return to the EXEC prompt. Check your X client system documentation for specific information about exiting an XRemote session.

To exit a SLIP and PPP, you must hang up the dial-in connection, usually with a command that your dial-in software supports.

To exit a LAT, Telnet, rlogin, TN3270, and X.3 PAD session begun from the router to a remote device, enter the escape sequence (**Ctrl-Shift-6** then **x** [**Ctrl^x**] by default) and enter the **disconnect** command at the EXEC prompt. You can also log off the remote system.

Except for XRemote, you also can escape to the EXEC prompt and enter either of the following commands to terminate an active terminal session:

**exit**  
**logout**

To exit a Telnet session *to* a router, see the following section, “Log Out of a Router.”

## Log Out of a Router

The method you use to 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 connected physically to the console port of the router, you can disconnect from the console port by physically disconnecting the cable from the console port of the router.
- 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 line through a synchronous interface on the router, you can issue any of the following commands to disconnect:

— **close**  
— **exit**  
— **logout**  
— **quit**

## Disconnect a Line

To disconnect a line, perform the following task in EXEC mode:

Task	Command
Disconnect a line.	<b>disconnect</b> [ <i>connection</i> ]

We recommend that you avoid disconnecting a line to end a session. Instead, log off the host to allow the router to clear the connection. Then end the session. Only if you cannot log out of an active session should you disconnect the line.

## Clear TCP/IP Connections

To clear a TCP connection, perform the following task in privileged EXEC mode:

Task	Command
Clear a TCP connection.	<b>clear tcp</b> { <b>line</b> <i>line-number</i>   <b>local</b> <i>host-name port</i>   <b>remote</b> <i>host-name port</i>   <b>tcb</b> <i>address</i> }

The **clear tcp** command is particularly useful for clearing hung TCP connections.

The **clear tcp line** *line-number* command terminates the TCP connection on the specified TTY line. Additionally, all TCP sessions initiated from that TTY line are terminated.

The **clear tcp local** *host-name port* **remote** *host-name port* command terminates the specific TCP connection identified by the host name/port pair of the local and remote router.

The **clear tcp tcb** *address* command terminates the specific TCP connection identified by the TCB address.

For examples of clearing TCP connections, see the “Clear TCP/IP Connection Examples” section in this chapter.

## Change Terminal Session Parameters

This section explains how to change terminal and line settings locally. The local settings temporarily override the settings made by the system administrator and remain in effect only until you exit the system.

You can configure the Cisco IOS software to save local parameters between sessions. These local parameters are set with the EXEC **terminal** commands listed in this section.

The following sections describe the more common local changes to the terminal and line settings:

- Select a Preferred Connection Protocol for a Session
- Set Communication Parameters for the Current Session
- Define Special Characters for the Current Session
- Specify Telnet Operation Characteristics
- Specify an International Character Display for the Current Session

The following sections describe the less common local changes to the terminal and line settings:

- Change Character Padding for the Current Session
- Change the Terminal and Keyboard Type
- Change the Terminal Screen Length and Width
- Change Packet Output Notification
- Change the Packet Dispatch Character for the Current Session
- Display Debug Messages on the Console and Terminals
- Change Flow Control for the Current Session

## Select a Preferred Connection Protocol for a Session

To specify the preferred protocol to use for the current session when a command does not specify one, perform the following task in EXEC mode:

Task	Command
Specify the protocol for the Cisco IOS software to use for the current session if the user did not specify a protocol.	<b>terminal transport preferred</b> {all   lat   mop   nasi   none   pad   rlogin   telnet   v120}

The preferred transport type is your preferred connection protocol. This setting specifies a protocol search order that the Cisco IOS software uses when it attempts to resolve a device name that you enter, but you do not specify a connection protocol. For example, if you want to connect to a TCP/IP host named host1 and want to use Telnet, you type **telnet host1**. However, if your preferred connection protocol is set to Telnet, you could type **host1** and be connected to the device. A host name might be valid for multiple protocols. If the address or service does not match the preferred protocol, all other valid connection protocols are searched to find a valid match for the name.

For router software images that support LAT, the default protocol for outgoing connections is LAT. For router software images that do not support LAT, the default protocol for outgoing connections is Telnet. For incoming connections, all the supported network protocols are accepted (the default protocol is **all**).

The Cisco IOS software accepts a host name entry at the EXEC prompt as a Telnet command. If you enter the host name incorrectly, the Cisco IOS software interprets the entry as an incorrect Telnet command and provides an error message indicating that the host does not exist. The **transport preferred none** command disables this option so that if you enter a command incorrectly at the EXEC prompt, the Cisco IOS software does not attempt to make a Telnet connection.

## Set Communication Parameters for the Current Session

The Cisco IOS software supplies the following default serial communication parameters for terminal and other serial device operation:

- 9600 bits per second (bps) line speed
- 8 data bits
- 2 stop bits
- No parity bit

You can change these parameters as necessary to meet the requirements of the terminal or host to which you are attached. To do so, perform one or more of the following tasks in EXEC mode:

Task	Command
Set the line speed for the current session. Choose from line speed, transmit speed, or receive speed.	<b>terminal speed</b> <i>bps</i> <b>terminal txspeed</b> <i>bps</i> <b>terminal rxspeed</b> <i>bps</i>
Set the data bits for the current session.	<b>terminal databits</b> {5   6   7   8}
Set the stop bits for the current session.	<b>terminal stopbits</b> {1   1.5   2}
Set the parity bit for the current session.	<b>terminal parity</b> {none   even   odd   space   mark}

You can also configure these parameters in line configuration mode for all connections to the line. Refer to “Configuring Terminal Lines and Modem Support” chapter to configure these parameters for more than the current session.

### Define Special Characters for the Current Session

You can modify key sequences to execute functions for system escape and terminal pause. To modify these sequences for the current session, perform one or more of the following tasks in EXEC mode:

Task	Command
Change the system escape sequence for the current session. The escape sequence indicates that the codes that follow have special meaning. The default sequence is Ctrl-^.	<b>terminal escape-character</b> <i>ASCII-number</i>
Define 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.	<b>terminal hold-character</b> <i>ASCII-number</i>

The **terminal escape-character** 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 you to EXEC mode when you are connected to another computer.

You can also configure these parameters in line configuration mode for all connections to the line. Refer to the “Define Escape Character and Other Key Sequences” section in the “Configuring Terminal Lines and Modem Support” chapter to configure these parameters for more than the current session.

### Specify Telnet Operation Characteristics

This section contains the following sections:

- Generate a Hardware Break Signal for a Reverse Telnet Connection.
- Set the Line to Refuse Full-Duplex, Remote Echo Connections.
- Allow Transmission Speed Negotiation.
- Synchronize the Break Signal.
- Change the End-of-Line Character.

#### Generate a Hardware Break Signal for a Reverse Telnet Connection

To cause the router 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, perform the following task in EXEC mode:

Task	Command
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.	<b>terminal telnet break-on-ip</b>

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.

---

**Note** This command applies only to access server products. It is not supported on standalone routers.

---

### Set 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, remote echo options on incoming connections, perform the following task in EXEC mode:

Task	Command
Set the current line to refuse to negotiate full-duplex for the current session.	<b>terminal telnet refuse-negotiations</b>

---

**Note** This command applies only to access server products. It is not supported on standalone routers.

---

### Allow Transmission Speed Negotiation

To allow the Cisco IOS software to negotiate transmission speed for the current line and session, perform the following task in EXEC mode:

Task	Command
Allow the Cisco IOS software to negotiate transmission speed for the current line and session.	<b>terminal telnet speed</b> <i>default-speed</i> <i>maximum-speed</i>

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.

---

**Note** This command applies only to access server products. It is not supported on standalone routers.

---

### Synchronize 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, perform the following task in EXEC mode:

Task	Command
Cause the Cisco IOS software to send a Telnet Synchronize signal when it receives a Telnet Break signal on the current line and session.	<b>terminal telnet sync-on-break</b>

**Note** This command applies only to access server products. It is not supported on standalone routers.

### Change the End-of-Line Character

The end of each line typed at the terminal is ended with a Return (CR). To cause the current terminal line to send a CR as a CR followed by a NULL instead of a CR followed by a line feed (LF), perform the following task in EXEC mode:

Task	Command
Cause the current terminal line to send a CR as a CR followed by a null instead of a CR followed by a line feed (LF).	<b>terminal telnet transparent</b>

This command ensures interoperability with different interpretations of end-of-line handling in the Telnet protocol specification.

**Note** This command applies only to access servers. It is not supported on standalone routers.

### Specify an International Character Display for the Current Session

To specify a character set based on hardware, software, or on a per-line basis, perform the following appropriate tasks in EXEC mode:

Task	Command
Set the number of data bits per character that are generated and interpreted by hardware for the current session.	<b>terminal databits {5   6   7   8}</b>
Set the number of data bits per character that are generated and interpreted by software for the current session.	<b>terminal data-character-bits {7   8}</b>
Specify the character set used in EXEC and configuration command characters on a per-line basis for the current session.	<b>terminal exec-character-bits {7   8}</b>
Specify the character set used in special characters (such as software flow control, hold, escape, and disconnect characters) on per-line basis for the current session.	<b>terminal special-character-bits {7   8}</b>

You can also configure these parameters in line configuration mode for all connections to the line. Refer to the “Configure Data Transparency” section in the “Configuring Terminal Lines and Modem Support” chapter earlier in this publication to configure these parameters for more than the current session.

## Change Character Padding for the Current Session

Character padding adds a number of null bytes to the end of the string and can be used to make a string an expected length for conformity. You can change the character padding on a specific output character. To change character padding on a specific output character for the current session, perform the following task in EXEC mode:

Task	Command
Set padding on a specific output character for the specified line for this session.	<b>terminal padding</b> <i>ASCII-number count</i>

## Change the Terminal and Keyboard Type

To specify the type of terminal connected to the current line for the current session, perform the following task in EXEC mode:

Task	Command
Specify the terminal type for this session.	<b>terminal terminal-type</b> <i>terminal-type</i>

Indicate the terminal type if it is different from the default of VT100. This default is used by TN3270 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, perform the following task in EXEC mode:

Task	Command
Specify the keyboard type for this session.	<b>terminal keymap-type</b> <i>keymap-name</i>

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 and give you their names.

You can also configure these parameters in line configuration mode for all connections to the line. Refer to the “Specify the Terminal and Keyboard Type” section in the “Configuring Terminal Lines and Modem Support” chapter earlier in this publication to configure these parameters for more than the current session.

## Change the Terminal Screen Length and Width

To set the number of lines on the current terminal screen for the current session or to set the number of character columns on the terminal screen for the current line, perform one of the following tasks in EXEC mode:

Task	Command
Set the screen length for the current session.	<b>terminal length</b> <i>screen-length</i>
Set the screen width for the current session.	<b>terminal width</b> <i>characters</i>

You can also configure these parameters in line configuration mode for all connections to the line. Refer to the “Set the Terminal Screen Length and Width” section in the “Configuring Terminal Lines and Modem Support” chapter earlier in this publication to configure these parameters for more than the current session.

### Change Packet Output Notification

You can set up a line to inform a user who has multiple, concurrent Telnet sessions when output is pending on a session other than the current one. To do so, perform the following task in EXEC mode:

Task	Command
Set up a line to notify a user of pending output for the current session.	<b>terminal notify</b>

You might want to know, for example, when another connection receives mail or a message.

To configure output notification for more than the current session, refer to the “Set Pending Output Notification” section in the “Configuring Terminal Lines and Modem Support” chapter earlier in this publication.

### Change the Packet Dispatch Character for the Current Session

To change the packet dispatch character for the current session, perform the following task in EXEC mode:

Task	Command
Define a character that triggers packet transmission for the current session.	<b>terminal dispatch-character</b> <i>ASCII-number1</i> [ <i>ASCII-number2 . . . ASCII-number</i> ]

To configure dispatch sequences and TCP state machines that transmit data packets upon receipt of the defined character or sequence of characters, refer to the “Create Character and Packet Dispatch Sequences” section in the “Configuring Terminal Lines and Modem Support” chapter earlier in this publication.

### Display Debug Messages on the Console and Terminals

To display **debug** command output and system error messages in EXEC mode on the current terminal, perform the following task in privileged EXEC mode:

Task	Command
Display debug command output and system error messages in EXEC mode on the current terminal.	<b>terminal monitor</b>

Remember that all terminal parameter-setting commands are set locally and do not remain in effect after a session is ended. You must perform this task at the privileged-level EXEC prompt at each session to see the debugging messages.

## Change Flow Control for the Current Session

To configure flow control between the router and attached device for this session, perform one of the following tasks in EXEC mode:

Task	Command
Set the terminal flow control for this session.	<b>terminal flowcontrol</b> { <b>none</b>   <b>software</b> [ <b>in</b>   <b>out</b> ]   <b>hardware</b> }
Set the flow control start character in the current session.	<b>terminal start-character</b> <i>ASCII-number</i> <sup>1</sup>
Set the flow control stop character in the current session.	<b>terminal stop-character</b> <i>ASCII-number</i> <sup>1</sup>

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 “Configure Flow Control” section in the “Configuring Terminal Lines and Modem Support” chapter earlier in this publication.

## Connection Examples

Use the examples in the following sections to understand how to create connections between network devices:

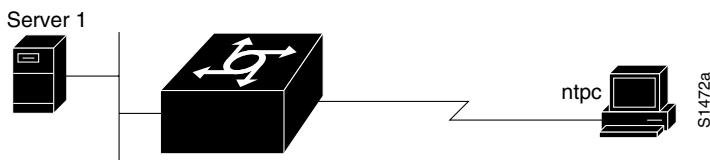
- PPP Connection Example
- SLIP Connection Examples
- XRemote Examples
- Telnet Connection Examples
- rlogin Example
- Switch between Telnet and rlogin Sessions Examples
- LAT Connection Examples
- Define a Group Code List for Outgoing LAT Connections Example
- TN3270 Connection Example
- Mobile Remote Node Example
- Protocol Translation Session Examples
- X.3 PAD Session Example
- Change a Login Name Example
- Specify a TACACS Host Example
- Set X.3 PAD Parameters Example
- Clear TCP/IP Connection Examples

## PPP Connection Example

The following example shows a line that is in asynchronous mode using PPP encapsulation (see Figure 22). The name of the PC is ntpc. Assuming that the name ntpc is in the Domain Naming System (DNS), the access server can match a real IP address. The PC must be running a terminal emulator program.

```
Router> ppp ntpc@server1
```

**Figure 22 Using the PPP EXEC Command**



## SLIP Connection Examples

The following example illustrates how to make a connection when the system administrator defines a default IP address by including the **peer default ip address** command in interface configuration mode.

---

**Note** The **peer default ip address** command replaces the **async default ip address** command.

---

Once a correct password is entered, you are placed in SLIP mode, and the IP address appears.

```
Router> slip
Password:
Entering SLIP mode.
Your IP address is 192.31.7.28, MTU is 1524 bytes
```

The following example illustrates the prompts displayed and the response required when dynamic addressing is used to assign the SLIP address:

```
Router> slip
IP address or hostname? 192.31.6.15
Password:
Entering SLIP mode
Your IP address is 192.31.6.15, MTU is 1524 bytes
```

In the preceding example, the address 192.31.6.15 has been assigned as the default. Password verification is still required before SLIP mode can be enabled.

```
Router> slip default
Password:
Entering SLIP mode
Your IP address is 192.31.6.15, MTU is 1524 bytes
```

The following example illustrates the implementation of header compression on the interface with the IP address 128.66.2.1:

```
Router> slip 128.66.2.1 /compressed
Password:
Entering SLIP mode.
```

```
Interface IP address is 128.66.2.1, MTU is 1500 bytes.
Header compression will match your system.
```

In the preceding example, the interface is configured for **ip tcp header-compression passive**, which permitted the user to enter the **/compressed** keyword at the EXEC mode prompt. The message “Header compression will match your system” indicates that the user specified compression. If the line was configured for **ip tcp header-compression on**, this line would read “Header compression is On.”

The following example specifies a TACACS server named *parlance* for address authentication:

```
Router> slip 1.0.0.1@parlance
Password:
Entering SLIP mode.
Interface IP address is 1.0.0.1, MTU is 1500 bytes
Header compression will match your system.
```

## XRemote Examples

Use the examples in this section to understand how to make XRemote connections. This section contains the following examples:

- Connect through Automatic Session Startup with XDMCP Server Example
- Connect through Automatic Session Startup with DECwindows Login via LAT Example
- Enable XRemote Manually Example
- Connect an X Display Terminal Example
- Make XRemote Connections between Servers Example

### Connect through Automatic Session Startup with XDMCP Server Example

The following example starts a session with a remote host named star:

```
Router> xremote xdm star
```

### Connect through Automatic Session Startup with DECwindows Login via LAT Example

The following example begins connection with a LAT service named WHIRL:

```
Router> xremote lat WHIRL
```

### Enable XRemote Manually Example

The following example illustrates how a successful manual XRemote session begins:

```
dialup> xremote
XRemote enabled; your display is dialup:2006
Start your clients and type XRemote again
```

The system replies with a message informing you of your X display location. Use this information to tell the host the location of your X display server.

If no clients are found, you see the following message:

```
No X clients waiting - check that your display is darkstar:2006
```

Check your hosts to determine whether an error has occurred when the session started. The most likely causes are that there is an improperly specified display location or the host computer did not recognize the name of your server.

### Connect an X Display Terminal Example

The following example shows how to make a connection from an X display terminal through a server to a host running client programs:

**Step 1** Enter the **xremote** command at the EXEC prompt.

```
dialup> xremote
```

**Step 2** Read and follow the instruction from the host.

```
XRemote enabled; your display is dialup:2006
Start your clients and type XRemote again
```

**Step 3** Connect to the client.

```
dialup> telnet eureka
Trying EUREKA.NOWHERE.COM (252.122.1.55)... Open

SunOS UNIX (eureka)
```

**Step 4** Log on at the prompt.

```
login: deal
Password:
Last login: Fri Apr 1 17:17:46 from dialup.nowhere.com
SunOS Release (SERVER+FDDI+DBE.patched) #14: Fri Apr 8 10:37:29 PDT 1994
```

**Step 5** At the client prompt, enter the display name from Step 3 in this procedure and the **xterm** command.

```
eureka% setenv DISPLAY dialup:2006
eureka% xterm &
[1] 15439
```

**Step 6** Disconnect from the client.

```
eureka% logout

[Connection to EUREKA closed by foreign host]
```

**Step 7** Begin the XRemote session.

```
dialup> xremote
Entering XRemote
```

The server and X terminal stay in XRemote mode until either the display manager terminates the session, or a reset request is received from the X terminal.

```
Connection closed by foreign host.
eureka%
```

## Make XRemote Connections between Servers Example

This section provides two examples of XRemote connections between servers.

The following example shows how an XRemote connection is established for a configuration such as the one shown in Figure 21 in the “Establish XRemote Sessions between Servers” section earlier in this chapter. This example assumes that the administrator has set the display environment variable to identify and match the user’s X display terminal.

**Step 1** From the PCX, MacX, or UNIX machine in Figure 21 on page 87, the user connects to port 9003 on Access Server 1. If your administrator has configured a rotary number 7, the user connects to port 10007. For more information about rotary groups, refer to “Configuring Terminal Lines and Modem Support” chapter in this publication and to the *Access Services Command Reference*.

Access Server 1 connects the user to a modem.

The modem calls Access Server 2.

**Step 2** Enter the **xremote** command at the Access Server 2 prompt.

**Step 3** Connect to the remote host from Access Server 2 using the **telnet** command.

**Step 4** Start the X client program that runs on the remote host and displays on the X display server (PCX, MacX, or UNIX host).

**Step 5** Escape from the remote host back to the Access Server 2, or log out if clients were run in the background, and enter the **xremote** command again at the Access Server 2 prompt.

The following example shows the steps to make an XRemote connection between servers. The number 9016 in the first line of the display indicates a connection to individual line 16. If the administrator had configured a rotary connection, the user would enter 10000 plus the number of the rotary (instead of 9016).

**Step 1** Enter the **telnet** command to make the connection.

```
space% telnet golden-road 9016
Trying 192.31.7.84 ...
Connected to golden-road.cisco.com.
Escape character is '^]'.
```

**Step 2** Supply the password for TACACS verification.

```
User Access Verification
```

```
Password:
Password OK
```

```
--- Outbound XRemote service ---
Enter X server name or IP address: innerspace
Enter display number [0]:
```

```
Connecting to tty16... please start up XRemote on the remote system
```

**Step 3** Dial in to the remote system using the modem, and then log on.

```
atdt 13125554141
DIALING
RING
CONNECT 14400
```

```
User Access Verification
```

```
Username: deal
```

```
Password:
```

```
Welcome to the cisco dial-up access server.
```

**Step 4** Enter the **xremote** command at the EXEC prompt, then follow the instructions from the host.

```
dialup> xremote
XRemote enabled; your display is dialup:2006
Start your clients and type XRemote again
```

**Step 5** Connect to the client.

```
dialup> telnet sparks
Trying SPARKS.NOWHERE.COM (252.122.1.55)... Open

SunOS UNIX (sparks)

login: deal
Password:
Last login: Fri Apr 1 17:17:46 from dialup.nowhere.com
SunOS Release (SERVER+FDDI+DBE.patched) #14: Fri Apr 8 10:37:29 PDT 1994
```

**Step 6** At the client prompt, enter the display name from step 4 and the **xterm** command.

```
sparks% setenv DISPLAY dialup:2006
sparks% xterm &
[1] 15439
```

**Step 7** Disconnect from the client.

```
sparks% logout

[Connection to SPARKS closed by foreign host]
```

**Step 8** Begin the XRemote session.

```
dialup> xremote
Entering XRemote
```

Once the connection is closed by the foreign host, the Xterm window appears on the local workstation screen.

```
Connection closed by foreign host.
sparks%
```

## Telnet Connection Examples

The following example routes packets from the source system host1 to kl.sri.com, then to 10.1.0.11, and finally back to host1:

```
Router> connect host1 /route:kl.sri.com 10.1.0.11 host1
```

The following example connects to a host with logical name host1:

```
Router> host1
```

## rlogin Example

The following example makes an rlogin connection to a host at address 108.33.21.2 and enables the message mode for debugging:

```
Router> rlogin 108.33.21.2 debug
```

## Switch between Telnet and rlogin Sessions Examples

The following example shows how to escape out of a connection to the host host1 and to resume connection 2:

```
host1% ^^X
Router> resume 2
```

You can omit the command name and simply type the connection number to resume that connection. The following example illustrates how to resume connection 3:

```
Router> 3
```

## LAT Connection Examples

The following example establishes a LAT connection from the router named Router to host eng2:

```
Router> lat eng2
Trying ENG2...Open
      ENG2 - VAX/VMS V5.2
Username: JSmith
Password:
      Welcome to VAX/VMS version V5.2 on node ENG2
      Last interactive login on Friday, 1-APR-1994 19:46
```

The system informs you of its progress by displaying the messages “Trying <system>...” and then “Open.” If the connection attempt is not successful, you receive a failure message.

The following example establishes a LAT connection from the router named Router to our-modems and specifies port 24, which is a special modem.

```
Router> lat our-modems port 24
```

The following example establishes a LAT connection from the router named Router to our-modems and specifies a node named eng:

```
Router> lat our-modems node eng
```

The following example uses the LAT session debugging capability:

```
Router> lat Eng2 /debug
Trying ENG2...Open
      ENG2 - VAX/VMS V5.2
Username: JSmith
Password:
      Welcome to VAX/VMS version V5.2 on node ENG2
      Last interactive login on Tuesday, 5-APR-1994 19:02
[Set Flow out off, Flow in on, Format 8:none, Speed 9600/9600]
[Set Flow out off, Flow in on, Format 8:none, Speed 9600/9600]
$ set ter/speed=2400
[Set Flow out off, Flow in on, Format 8:none, Speed 2400/2400]
```

A variety of LAT events are reported, including all requests by the remote system to set local line parameters. The messages within brackets ([ ]) are the messages produced by the remote system setting the line characteristics as the operating system defaults.

## Define a Group Code List for Outgoing LAT Connections Example

The following example defines a group code list for the outgoing group 4 LAT connection:

```
Router> terminal lat out-group 4, 6-189
```

## TN3270 Connection Example

The following example establishes a terminal session with an IBM host named finance:

```
Router> tn3270 finance
```

To terminate an active TN3270 session, log off the remote system by issuing the command specific to that system (such as **exit**, **logout**, **quit**, or **close**). You can also enter the escape sequence (**Ctrl-Shift-6** then **x** [**Ctrl^x**] by default) and enter the **disconnect** command at the EXEC prompt. Because the **disconnect** command can “hang” a port, we recommend that you avoid using it routinely when you exit a session.

## Mobile Remote Node Example

The following example establishes a network layer connection with an IBM host named mktg:

```
Router> tunnel mktg
```

## Protocol Translation Session Examples

This section illustrates how to make connections for protocol translation using the two-step and one-step protocol translation methods.

### Using One-Step Protocol Translation for TCP (Telnet) to X.25 Host Connections Example

The following is an example of one-step protocol translation. A user at a UNIX workstation makes a connection to a remote X.25 host named host1 over an X.25 PDN. The router automatically converts the Telnet connection request to an X.25 connection request and transmits it as specified in the system configuration.

**Step 1** Establish a connection by entering the **telnet** EXEC command at the UNIX workstation system prompt.

```
unix% telnet host1
```

---

**Note** This example assumes that the name host1 is known to the UNIX host (obtained using DNS, IEN116 or a static table) and is mapped to the IP address used in a **translate** command. (See the *Access Services Command Reference*.)

---

The router accepts the Telnet connection and immediately forms an outgoing connection with remote host1 as defined in a **translate** command in your router's active configuration file. The device host1 sets several X.3 parameters, including local echo. Because the Telnet connection is already set to local echo (at the UNIX host), no changes are made on the TCP connection.

**Step 2** Enter a username, then a password when the host1 connection prompts for them. The router converts this to a Telnet option request on the UNIX host, which then stops the local echo mode.

At this point you are connected to the PAD application that sets the X.3 PAD parameters (although they can always be overridden using the **resume** or **x3** commands).

**Step 3** When you are finished with the connection, enter the escape sequence (**Ctrl-Shift-6** then **x** [**Ctrl^x**] by default) to exit back to the host connection, then enter the appropriate command to close the connection.

The device host1 immediately closes the X.25 connection. The router then drops the TCP connection, leaving you at the UNIX system prompt.

## Using Two-Step Protocol Translation for TCP-to-PAD Connections Examples

The following example shows a connection from a local UNIX host (host1) to a router (router1) as the first step in a two-step translation process:

```
host1% telnet router1
```

The following example shows a connection from router1 to a host named ibm3278 as the second step in a two-step translation process:

```
Tasmania> tn3270 ibm3278
ibm3278%
```

In the following example, you connect directly from a terminal or workstation on a TCP/IP network to a router, and then to a database called Information Place on an X.25 packet data network. The database has a service address of 71330.

**Step 1** Make the following connection requests at a UNIX workstation as a first step to logging into the database Information Place.

```
unix% telnet router1
```

If the router named router1 is accessible, it returns a login message and you enter your login name and password.

**Step 2** Connect from the router to the database Information Place, which is on an X.25 host. You connect to an X.25 host using the **pad EXEC** command followed by the service address.

```
router1> pad 71330
```

Once the connection is established, the router immediately sets the PAD to single-character mode with local echoing, because these are the settings that the router expects. The PAD responds with its login messages and a prompt for a password.

```
Trying 71330...Open
Welcome to the Information Place
Password:
```

Because the password should not echo on your terminal, the PAD requests remote echoing so that characters will be exchanged between the PAD and the router, but not echoed locally or displayed. After the password is verified, the PAD again requests local echoing from the router.

**Step 3** Complete this sample session by logging off, which returns you to the router system EXEC prompt.

**Step 4** Execute the **quit EXEC** command, and the router drops the network connection to the PAD.

## Changing Parameters and Settings Dynamically Example

The following example shows how to change parameters during a session. In this example you must edit information on a remote host and change the X.3 PAD parameters that define the editing characters from the default Delete key setting to the **Ctrl-D** sequence. (Refer to the “ASCII Character Set” appendix of the *Configuration Fundamentals Command Reference* for a list of ASCII characters.)

**Step 1** Enter the escape sequence to return to the system EXEC prompt.

```
Ctrl1^x
```

**Step 2** Enter the **resume** command with the **/set** keyword and the desired X.3 parameters.

X.3 parameter 16 sets the Delete function. ASCII character 4 is the Ctrl-D sequence.

```
Tasmania> resume /set 16:4
```

The session resumes with the new settings. If the information is not displayed correctly, you can set the **/debug** switch to check that your parameter setting has not been changed by the host PAD.

**Step 3** Enter the escape sequence to return to the system EXEC prompt, then enter the **resume** command with the **/debug** switch. The **/debug** switch provides helpful information about the connection.

```
Tasmania> resume /debug
```

You also can set a packet dispatch character or sequence using the **terminal dispatch-character** command, as shown in the following example:

**Step 1** Set the ESC key (ASCII character 27) to a dispatch character by entering the following command:

```
Tasmania> terminal dispatch-character 27
```

**Step 2** Enter the following command to return to the PAD connection:

```
Tasmania> resume
```

The ESC key is set to a dispatch character, and the original PAD connection is resumed.

## X.3 PAD Session Example

The following example starts a PAD session:

```
Router>pad 123456789
Trying 123456789...Open
Router2>
```

## Change a Login Name Example

The following example shows how login usernames and passwords can be changed. In this case, a user currently logged on 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 EXEC at the user-level.

```
Router> login
Username: user2
Password:
Router>
```

## Specify a TACACS Host Example

In the following example, user1 specifies the TACACS host host1 to authenticate the password:

```
george> login
Username: user1@host1
Translating "HOST1"...domain server (131.108.1.111) [OK]
```

## Set X.3 PAD Parameters Example

The following example illustrates how to reset the outgoing connection default for local echo mode on a router. The `/set` switch sets the X.3 parameters defined by parameter number and value, separated by a colon.

```
router> resume 3 /set 2:1
```

## Clear TCP/IP Connection Examples

This section contains examples for clearing TCP connections. For task information about clearing TCP connections, refer to the “Clear TCP/IP Connections” section in this chapter.

### TTY Line Number Example

The following example clears a TCP connection using its TTY line number. The `show tcp` command displays the line number (tty2) that is used in the `clear tcp` command.

```
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, KRRT: 0 ms  
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 300 ms
```

```
Router# clear tcp line 2  
[confirm]  
[OK]
```

### Host Name and Port Pairs Example

The following example clears a TCP connection by specifying its local router host name and port and its remote router host name and port. The **show tcp brief** command displays the local (Local Address) and remote (Foreign Address) host names and ports to use in the **clear tcp** 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]
```

### TCB Address Example

The following example clears a TCP connection using its TCB address. The **show tcp brief** command displays the TCB address to use in the **clear tcp** 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]
```