

Chapter 3

Using the Terminal Server

3

This chapter provides an introduction to using Cisco terminal servers. The emphasis here is on the various elements of a Cisco terminal server's user interface. The following topics are addressed:

- Basic discussion of terminal connection service features
- Running the system and accessing different operational modes
- Procedures for making rudimentary connections
- An outline of common user connection service tools available with all protocols supported by Cisco's terminal servers

Note: This chapter is a survey of Cisco's terminal server connection capabilities. The objective is to provide users with the essentials for making basic connections. Other chapters provide more depth in terms of configuration and customization.

Please refer to the "First-Time Startup" chapter for specifics concerning system configuration using the **setup** command facility. Refer to the "Configuring the System" chapter for details about system-wide configuration features. The "Managing the System" chapter addresses facilities for monitoring and managing Cisco terminal servers. Configuration utilities associated with specific transmission protocols (TCP/IP, DEC LAT, SLIP, IBM TN3270, X.25, and XRemote) are addressed in the "Transmission Protocols" section of this guide. If you have special applications, or wish to customize your connections, please refer to these other chapters for details.

Using the EXEC Command Interpreter

Before exploring the specific connections commands provided with Cisco terminal servers, you should take time to familiarize yourself with the basic command line user interface integral to all Cisco products. This section introduces you to the basic Cisco terminal server user interface environment. This environment is generally referred to as the EXEC Command Interpreter (or EXEC).

The EXEC interprets the commands you type and carries out the corresponding operations. You can type commands when you see the system prompt, which is the host name followed by an angle bracket (>).

Command Use

You may type commands in uppercase, lowercase, or both uppercase and lowercase letters. You may also abbreviate commands and other keywords to the number of characters that cause the command to be a unique abbreviation. For example, you can abbreviate the **show** command to **sh**.

If you make a typing mistake, you can erase characters with the Delete or the Backspace key. Press either key to erase the last character you typed. To erase the entire line, type Ctrl-U. (This notation means “Hold down the Ctrl key and press the U key.”)

The terminal server acts on most commands after you press the Return key. You can abort a command at any time by typing Ctrl-C.

Note: All standard Cisco Ctrl-key conventions are described in the “Configuring the System” chapter.

To list available EXEC commands, type a question mark (?). You can often enter a question mark (help command) to obtain more information about commands. For example, type `terminal ?` to obtain a list of **terminal** commands or `show ?` to obtain a list of **show** commands.

Certain EXEC commands produce multiple screens of output. At the end of each screen, the EXEC pauses and displays:

-More-

Type a space to continue the output; type anything else to return to the system command prompt.

Command Levels

For security purposes, the EXEC has two levels of access: *user* and *privileged*. The commands available at the user level are a subset of the commands available at the privileged level. Because many of the privileged commands set terminal server operating parameters, the privileged level should be password-protected to prevent its unauthorized use. The system prompt for the privileged level ends with a pound sign (#) instead of an angle bracket (>).

The **enable** command allows access to the privileged level, prompting for a password if one has been set with the **enable-password** configuration command. For more information, see the “Establishing Passwords and System Security” section in the “Configuring the System” chapter of this manual. To list available EXEC commands, use the ? (question mark) command. At the user level, the ? command produces this list:

```

ts>?
connect <host>      Connect to host - same as typing just a host name
disconnect <cn>    Break the connection specified by name or number
exit, quit         Exit from the EXEC
lat <service>      Connect to service using DEC LAT protocol
lock               Lock the terminal
login             Log in as a particular user
name-connection   Give a connection a logical name
resume           Make the named connection be current
rlogin <host>     Connect to host using rlogin protocol
show <cmd>        Information commands, type "show ?" for list
systat           Show terminal lines and users
telnet <host>     Connect to host using telnet protocol
tn3270 <host>     Connect to host using telnet protocol (3270)
terminal         Change terminal's parameters, type "terminal ?"
where            Show open connections
xremote          Enter xremote mode
<cr>            To resume connection

```

At the privileged level, the ? command lists the full terminal server command set:

```

ts#?
clear             Reinitialization functions, type "clear ?" for list
configure        Configure from terminal or over network
connect <host>   Connect to host - same as typing just a host name
debug            Enable debugging functions, type "debug ?" for list
disable          Turn off privileged commands
disconnect <cn> Break the connection specified by name or number
enable           Turn on privileged commands
exit, quit       Exit from the EXEC
lat <service>    Connect to service using DEC LAT protocol
lock            Lock the terminal
login           Log in as a particular user
name-connection Give a connection a logical name
ping            Send echo messages
reload          Halt and reload system
resume         Make the named connection be current
rlogin <host>   Connect to host using rlogin protocol
send <line>|*   Send message to a terminal line or lines
setup          Initialize system configuration
show <cmd>      Information commands, type "show ?" for list
systat         Show terminal lines and users
telnet <host>   Connect to host using telnet protocol
tn3270 <host>   Connect to host using telnet protocol (3270)
terminal       Change terminal's parameters, type "terminal ?"
test           Run hardware tests, type "test ?"
trace <address> Trace route to <address>
undebug        Disable debugging functions, type "undebug ?" for list
where          Show open connections
write          Write configuration memory, type "write ?" for list
xremote        Enter xremote mode
<cr>          To resume connection

```

Making Connections with a Cisco Terminal Server

The Cisco terminal server provides excellent depth of feature content and flexibility of connection service capabilities. Users can make simple automated terminal-to-host connections, or can establish custom terminal sessions tailored to specific terminal or application requirements. The following connection descriptions define the basic connection capabilities available to users and the syntax associated with each user level EXEC connection command. Refer to the chapters in the “Transmission Protocols” section of this manual, for details about protocol-related customization of terminal characteristics. The commands outlined here represent the basic connection services provided by Cisco terminal servers for DEC LAT, Telnet, Rlogin, TN3270, and XRemote.

Note: In many instances, extensive optional argument strings can be concatenated to the basic commands. In general, these options, and supporting configuration capabilities, are reserved for discussion in the protocol-specific chapters of this manual.

LAT Connections

DEC’s Local Area Transport (LAT) protocol is the protocol used most often to connect terminal servers to DEC hosts. LAT is a DEC-proprietary protocol. Cisco uses LAT technology licensed from DEC.

The LAT protocol is similar to TCP/IP’s Telnet protocol in that it allows a user at one site to establish a connection to a host at another site, then passes the keystrokes from one system to the other. A user can establish a LAT connection through the terminal server to a DEC host, simply by entering the host name.

Establishing a LAT Connection

Use the EXEC command **lat** to establish a connection to a LAT learned service. The command has this syntax:

```
lat name [node nodename | port portname | /debug]
```

Enter this command at the EXEC prompt. It can be entered simply by typing the name of the service to which you want to attach, or with the optional keywords listed in the above syntax description. You can also just simply enter the service name to make the connection.

The optional keyword **node** specifies connection to a specific LAT node which offers a service. Enter the name of the node in place of the *nodename* argument. If you do not include the node option, the node with the highest rating offering the service is used. Use the EXEC command **show lat nodes** to display information about all known LAT nodes (see the section “Monitoring LAT” in the “Configuring LAT” chapter for more information about this command).

The optional keyword **port** specifies a destination LAT port name. This keyword is ignored in most timesharing systems, but is used by terminal servers offering *reverse LAT* services. *Reverse LAT* services are discussed in the section “Inbound Session Support,” in the “Configuring LAT” chapter. Generally, *reverse LAT* involves connecting to one terminal server from another. In this case, the target terminal server runs the host portion of the protocol. Enter the port name in the format of the remote system in place of the *protamine* argument.

The optional keyword **/debug** is a switch that, when enabled, prints parameter changes and other special messages on the terminal.

Note: With Cisco’s implementation of LAT, you are not required to enter the word **lat** to establish a LAT connection. If you prefer, you can just enter the LAT learned service name. To show a listing of the available LAT learned services, use the **show lat services** command (see the section “Monitoring LAT” in the “Configuring LAT” chapter, for more information about this command).

Example:

The following example establishes a LAT connection from the terminal server named *ts* to the host *Eng2*.

```
ts>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, 6-APR-1990 19:46
```

The LAT protocol is explicitly specified in this example. You specify the protocol when your preferred transport is set to “none” or to another protocol. The terminal server responds with “Trying <system>...” and then “Open.” If the connection were not successful you would receive a failure message. Refer to the “Configuring the System” chapter for more information about setting terminal preferences.

SLIP Connections

The Serial Line Internet Protocol, or SLIP, defines a method of sending IP packets over standard RS-232 asynchronous serial lines.

SLIP is an inexpensive way of connecting PCs to a network. SLIP can be used over asynchronous dial-up modems, allowing computers in people’s homes to be connected to a network without the cost of a leased line. Dial-up SLIP links can also be used for remote sites that need only occasional or backup connectivity.

The terminal server can support any combination of SLIP lines and lines dedicated to normal asynchronous devices, such as terminals and modems.

The EXEC command **slip** begins a typical SLIP connection. Enter this command at the EXEC prompt:

slip

The terminal server displays in text form the address it expects the SLIP client to have, which enables SLIP mode. The SLIP client may also determine this address using the Boot Protocol (BootP).

Sample Session:

In this session, the user enters the slip EXEC command to make connection, and then the IP address to use is displayed:

```
ts>slip
Ip address: 192.31.7.28
```

Once the IP address and any login requirements from the host system (login name, password, and so on) are entered, SLIP confirms that SLIP mode has been successfully entered:

```
Entering SLIP mode...
```

Note: There are other ways to “start” SLIP connections. For instance, if the line configuration subcommand **slip dedicated** is included in the terminal server configuration, users will not be prompted with anything. Use the line configuration subcommand **slip interactive** to reinstate normal SLIP connection mode.

The chapter “Configuring SLIP,” describes how to configure a line for SLIP mode, and how to maintain the SLIP line.

Rlogin Connection

The **rlogin** command starts a connection using the UNIX rlogin protocol. The full syntax of the command follows:

rlogin *host* [**debug**]

The argument *host* is a host name or an Internet address. The optional keyword **debug** enables debugging output from the rlogin protocol.

An alternative to the Telnet protocol, the rlogin protocol provides superior flow control and output suppression. The Cisco implementation of rlogin does not subscribe to the “trusted” host model. That is, a user cannot automatically log onto a UNIX system from the terminal server, but must provide a user ID and a password for each connection.

For more information concerning rlogin connections, please refer to the “Configuring TCP/IP” chapter later in this manual.

Telnet Connections

The following section summarizes Telnet-based connection capabilities of Cisco terminal servers. For more information concerning Telnet connections, please refer to the “Configuring TCP/IP” chapter later in this manual.

Starting a Telnet Connection

Use the EXEC command **connect** or **telnet** to start a Telnet connection. The command has this syntax:

```
{connect | telnet} host [port] [/keyword]
```

The argument *host* is a host name or an Internet address. The optional argument *port* is a decimal TCP port number; the default is the Telnet server port (decimal 23) on the host. The optional argument *keyword* is one of the following:

- **/route:** *path*—Specifies loose source routing. The argument *path* is a list of host names or Internet addresses that specify network nodes, ending with the final destination. For example, the following command routes packets from the source system to *kl.sri.com*, then to 10.1.0.11, and finally to *mathom*:

```
ts>connect mathom /route:kl.sri.com 10.1.0.11 mathom
```

Note: The STS-10x terminal server does not support the **/route:** *path* keyword.

- **/line**—Enables Telnet line mode. In this mode, the terminal server does not send any data to the host until the user presses Return. The user can edit the line using the standard terminal server command editing characters (Backspace, Delete, Ctrl-U, Ctrl-W). The **/line** keyword is a local switch; the remote server is not notified of the mode change.
- **/debug**—Enables Telnet debugging mode.
- **/stream**—Turns on “stream” processing, which enables a raw TCP stream with no Telnet control sequences. A stream connection does no processing of Telnet options, and may be appropriate for connections to ports running UUCP and other non-Telnet protocols.

Note: With Cisco’s implementation of TCP/IP, you are not required to enter the word **connect** or **telnet** to establish a Telnet connection. If you prefer, you can just enter the learned host name. To show a listing of the available hosts, use the **show hosts** command (see the section “Monitoring TCP/IP” in the Configuring TCP/IP chapter for more information about this command).

You may omit the command word **connect** or **telnet**, if the host name you want to use is not the same as a terminal server command word.

The terminal server assigns a logical name to each connection; 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 EXEC command **name-connection**. If the name is already in use, the terminal server assigns a null name to the connection.

TN3270 Connections

If your network administrator has configured an appropriate default terminal type, then using the EXEC command **tn3270** is all you need to type to start a tn3270 session via a Cisco terminal server. The syntax for this command is as follows:

```
tn3270 hostname
```

The argument *hostname* is the name of a specific host on a network that is reachable by the terminal server. The default terminal emulation mode allows access using a VT100 emulation. If your terminal requires a different emulation, you must configure your terminal server to support different terminal types.

The following example use of the EXEC command **tn3270** results in the Cisco terminal server attempting to establish a terminal session with an IBM host named *finance*.

Example:

```
ts>tn3270 finance
```

Note: Unlike Telnet and LAT connections, you *must* enter the command **tn3270** in order to make a connection to a host with this command.

Establishing connections using alternative configurations (which must be included in the Cisco terminal server configuration) is explained in the next section. The process for configuring the Cisco terminal server to include alternative (and custom) terminal emulations is outlined in the “Configuring TN3270” chapter later in this manual.

XRemote Connections

If your host computer includes a server for XDMCP, such as the *xdm* program included in X11R4, you can use automatic session startup to initiate an XRemote session on an NCD X terminal.

The EXEC command used to initiate XRemote using XDMCP is **xremote xdm**. This command has the following syntax:

```
xremote xdm hostname
```


This command causes an XDMCP session start-up request to be made to a specified computer (*hostname*). If a host name is not specified, a broadcast message is sent to all hosts. The first host to respond by starting up a session is used.

The terminal 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.

Note: If your host does not include a server for XDMCP, you must perform the manual startup procedures outlined in the “Configuring XRemote” chapter later in this manual.

X.25 Terminal Server Support

Although Cisco terminal servers do not support PAD-based connection service for X.25 connection service, they do support for the use of X.25 as a transport mechanism for IP-based traffic. As you would expect, this capability covers Telnet, connect and rlogin connection commands, but also extends to XRemote and TN3270. In addition, a serial terminal server can provide SLIP on an asynchronous line, and route IP over X.25. LAT is not supported over X.25. Refer to the “Configuring X.25” chapter for complete details concerning configuration requirements for setting up X.25 encapsulation on terminal servers.

This capability is most useful on terminal servers with a serial network interface. From a user’s point of view, connections are relatively transparent when making a remote connection over a PDN. In general, users simply make a connection request.

Example X.25 Connection Command:

```
ts>telnet far_host
```

If properly configured, the terminal server takes this connection request, encapsulates the IP-based frame in an X.25 packet and sends it over the X.25 PDN to a specified location (based on the X.121 address). At the remote location, a Cisco router takes the request, de-encapsulates the X.25 packet, and puts the IP-based frame on the appropriate network. If all goes well, a virtual terminal connection is then made between the originating node and the target host.

Note: As an alternative, a Cisco Protocol Translator can be used to support this requirement via X.25 PAD capabilities. Refer to the *Cisco Protocol Translator Configuration and Reference* publication for details about support of X.25 connections with the protocol translator.

Common User Level Commands

The following user level commands and facilities are applicable to all transmission protocols supported by Cisco terminal servers:

- The escape sequence capability
- **resume** command
- **login** command
- **lock** command
- **where** command
- **name-connection** command
- **disconnect** command
- **shows session** command
- **show terminal** command
- **systat all** command
- **exit** and **quit** commands

Establishing Multiple Connections

You can have several concurrent connections open and switch back and forth between them. The number of connections that may be opened is defined by the **session-limit** command. Refer to the chapter “Configuring the System.” To switch between connections, first type the escape sequence, which by default is Ctrl-^, X (press the Ctrl, SHIFT and ^ key simultaneously, let go, then press the X key), to return to the system command prompt.

To make a new connection, use the procedure described in the paragraphs above. To get back to an existing connection, use the **resume** command, described next. Use the **where** command to list your open connections.

Switching Between Connections

The EXEC **resume** command returns to a previous connection. The general form is available with all connection protocols supported by the terminal server. The general command syntax is as follows:

```
resume [connection]
```

The optional argument *connection* is the connection name or number; the default is the most recent connection.

Note: Refer to the “Transmission Protocols” section for more information about **resume** command options available for specific protocols.

You can omit the command word **resume** and simply type the connection number to resume a connection. You can also return to the most recent session by just pressing the Return key.

Logging in as a Specific User

Use the EXEC command **login** to log in to a system with a specific user name. This command can be used to change your login name. One reason to change your login name is that an outgoing access list may depend on the specification of a particular user name. Refer to the discussion regarding Cisco's TACACS (and other user authentication) facilities in the "Configuring the System" chapter. In addition, IP access lists are discussed in the "Configuring TCP/IP" chapter. The syntax for the **login** command is:

login

The system returns prompts for user name and password. If both are entered correctly, your session becomes associated with the specified user name. If there is no match, your connection to the terminal server reverts to the user name with which the **login** command attempt was made, if applicable. If no login name and password were originally required, the connection reverts to a session that is not associated with any name.

Example:

In the example below a user named *foouser* wants to change the login name being used to *sloan*. After entering the **login** command, the new name is entered along with an incorrect password. The system rejects the attempt to change user name. Next *foouser* attempts to change the login name to *klaus*. This time the correct password is entered and the user is now allowed access at the user level prompt under the user name of *klaus*.

```
sys-prompt>login

Username:sloan
Password:
% Access denied
Still logged in as "foouser"
sys-prompt>login

Username:klaus
Password:
sys-prompt>
```

Locking the Terminal

If you want to prevent access to your terminal server session while keeping your connection sessions online, use the **lock** EXEC command. This command has the following syntax:

lock

The effect of this command is to lock your keyboard. It requires that the global configuration command **lockable** be included in your system configuration. (described in the chapter, "Configuring the System"). The **lock** EXEC command remains in effect until the **clear line**

privileged EXEC command (described in the chapter “Managing the System” “Transmission Protocols”) is executed. If you are able to lock your session, you will be prompted to specify a password, which can be any arbitrary string. Enter the password you want to assign. The screen clears and displays the message “Locked.” To regain access to your sessions, reenter the password. The terminal server honors session timeouts on a locked line.

Listing Open Connections

The EXEC command **where** displays information about all open LAT, Telnet, or rlogin connections associated with the current terminal line. The command has this simple syntax:

where

Following is a sample display:

```
ts>where
Conn Host          Address          Byte  Idle Conn Name
  1 MIS1            131.108.19.50   0     0   MIS1
*  2 OTTER          192.31.7.24     0     0   OTTER
```

The information it displays includes the host name, address, number of characters waiting to be sent to the terminal, idle time, and connection name. An asterisk (*) indicates the current connection.

Naming a Connection

The **name-connection** command assigns a logical name to a connection. The EXEC prompts for the connection number and name to assign when you enter this command. The **where** command displays a list of the assigned logical connection names.

The following example checks the connection number for the host *Eng1*, assigns the logical name *my host* to it, and then confirms the assignment.

```
ts>where
Conn Host          Address          Byte  Idle Conn Name
*  1 Eng1            192.31.6.22     0     0   Eng1
  2 Term2           192.33.6.21     0     0   Term2

ts>name-connection
Connection number:1
Enter logical name:  my host
Connection 1 to Eng1 will be named "my host" [confirm]
ts>where
Conn Host          Address          Byte  Idle Conn Name
*  1 Eng1            192.31.6.22     0     0   my host
  2 Term2           192.33.6.21     0     0   Term2
```

Aborting a Connection

The EXEC command **disconnect** aborts a connection, and has this syntax:

disconnect [*connection*]

The optional argument *connection* is a connection name or number; the default is the current connection. To list your open connection, use the **where** command.

```
ts>where
Conn Host      Address      Byte  Idle  Conn Name
*  1  Eng1      192.31.6.22   0    0  my host
   2  Term2     192.33.6.21   0    0  Term2

ts>disconnect my host
Closing connection to Eng1 [confirm]
```

Note: Do not use the **disconnect** command to end a session. Instead, log off the host, thus allowing the host to initiate the disconnect. If you cannot log off the host, use the **disconnect** command.

Displaying User Sessions

To display information about your active terminal sessions, use the **show sessions** EXEC command. This command has the following syntax:

show sessions

Example:

```
Conn Host      Address      Byte  Idle  Conn Name
*  1  GUN       131.222.3.11   0    0  GUN
   2  BIG       131.222.3.14   0    5  BIG
```

The information it displays includes the host name, address, number of characters waiting to be sent to the terminal, idle time, and connection name. An asterisk (*) indicates the user's current session.

Displaying Current Terminal Parameters

The EXEC command **show terminal** displays the configuration parameter settings for the current terminal line. Enter this command at the EXEC prompt:

show terminal

Following is a sample display of the command's output:

```
ts>show terminal
Line 3, Location: "Library x1234", Type: ""
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 9600/9600, no parity, 2 stopbits, 8 databits
The escape character is "^", followed by "x"
The local hold character is disabled
No flowcontrol in effect.
Status: Ready, Active
Capabilities: none
Idle EXEC timeout is 10 minutes.
Idle session timeout is not set.
Modem answer timeout is 15 seconds
Dispatch timeout is not set.
Allowed transports are telnet rlogin. Preferred is telnet
Disconnect character is not set
Activation character is ^M (13)
No output characters are padded
Characters causing immediate data dispatching:
Char   ASCII
```

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

Displaying Line Information

The EXEC command **systat** displays information about the active ports of the terminal server, and has this syntax:

systat [all]

A sample display follows:

```
ts>systat all
  Line  User      Host(s)      Idle  Location
  0 con 0
  1 tty 1
  2 tty 2      DAVE-SS2    1:32  Dave x1235
 * 3 tty 3      STUFT       0     Denise x1236
  4 tty 4      incoming    11:17 remote.host.com
  5 tty 5
  6 tty 6
  7 tty 7
 10 tty 10     STUFT       3     Mark x1237
 11 tty 11     HEAT       1     Eng32-1
 12 tty 12     STFUT     17:32  CD8 ???
 13 tty 13
 14 tty 14     MEDDLE     23    Marsha x1239
 15 tty 15
 16 tty 16
 17 tty 17
 20 tty 20
 21 tty 21
 22 tty 22
```

The information it displays includes the line number, connection name, idle time, and terminal location. The optional keyword **all** requests information about both active and inactive ports.

Exiting a Session

The **exit** or **quit** command terminates the EXEC command parser and closes any active terminal sessions. Enter either command when you are finished with all sessions.

