CHAPTER 4

Configuring Terminal Lines and Modem Support

This chapter explains how to configure the communication server for line, terminal, and modem connections. For a complete description of the commands mentioned in this chapter, refer to the "Terminal Line and Modem Support Commands" chapter in the *Access and Communication Servers Command Reference* publication.

For more information about making connections to network hosts through your communication server, refer to the *Cisco Access Connection Guide*.

Your system software permits you to connect to asynchronous serial devices such as terminals and modems, and to configure custom device operation. You can configure a single line or a range of lines. For example, you can configure one line for a laser printer and then configure a set of lines to switch incoming modem connections to the next available line. You can also customize your configurations. For example, you can define line-specific transport protocols, control character and packet transmissions, and establish time limits for user access.

Line Configuration Task List

The following sections describe line configuration tasks. One of the first things you will want to do for line configuration is to configure the lines for the terminals or other asynchronous serial devices attached to them. The line configuration tasks are listed next. Which tasks you perform and the order in which you perform them are determined entirely by the needs of your network environment.

- Prepare to Configure the Lines
- Configure Virtual Asynchronous Interfaces
- Define Terminal Operation Characteristics
- Record the Device Location
- Configure Software Flow Control
- Configure Software Flow Control Start and End Characters
- Define a Command String for Automatic Execution
- Configure Automatic Protocol Startup
- Create Character and Packet Dispatch Sequences
- Enable the Auxiliary Port
- Specify the Transport Protocol for a Specific Line
- Set a Line as Insecure
- Save Local Settings between Sessions

- Set Pending Output Notification
- Establish and Control the EXEC Process
- Configure Modem Lines
- Configure High-Speed Modem Support
- Connections to an Individual Line
- LPD Protocol Support
- Configure Rotary Groups
- Enable Password Checking at Login
- Support Reverse Telnet Connections
- Disable Enhanced Editing Mode
- Set a Terminal-Locking Mechanism
- Dedicate a Line to a Particular User
- Provide Line Connection Information after the Login Prompt
- Set a Line to Act as a Pipe for File Transfers
- Configure Chat Scripts for Asynchronous Lines
- Display Terminal Banner Messages
- Line Configuration Examples

See the end of this chapter for configuration examples. See the *Access and Communication Servers Command Reference* for information about the commands listed in this chapter.

Prepare to Configure the Lines

To configure a line, complete the following tasks:

Task		Command
Step 1	At the privileged EXEC prompt, enter configuration mode from the terminal.	**configure**[1][terminal]
Step 2	From global configuration mode, begin to configure a line.	**line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]
Step 3	Enter commands listed in this chapter to configure the line.	Use the commands listed in this chapter.
Step 4	Exit line configuration mode and return to EXEC mode.	**Ctrl-Z**
Step 5	Save the configuration changes to NVRAM.	**write memory**[1]

[1] These commands are documented in the “System Image and Configuration File Load Commands” chapter in the *Access and Communication Servers Command Reference* publication.

Once a line is configured, you can check its status by entering the EXEC **show users all** command.

Create Additional Virtual Terminal Lines

The **line vty** command accepts a *line number* larger than 5 (up to the maximum allowed on that communication server with its current configuration). The communication server dynamically creates all of the new virtual terminal lines between the current highest-numbered line and the number you specify. You can then configure those lines with additional line configuration commands.

See the section “Creating Additional Virtual Terminal Lines Example” at the end of this chapter for an example of how to add virtual terminal lines.

Eliminate Virtual Terminal Lines

To delete virtual terminal lines, perform the following task:

Task		Command
Step 1	At the privileged EXEC prompt, enter configuration mode from the terminal.	**configure**[1] **terminal**
Step 2	From global configuration mode, delete virtual terminal lines.	**no line vty** *line-number*

[1] This command is documented in the “System Image and Configuration File Load Commands” chapter in the *Access and Communication Servers Command Reference* publication.

The communication server deletes the *line number* and all virtual terminal line numbers above that number.

You cannot delete virtual terminal lines that are in use; attempting to do so results in a warning message. Virtual terminal lines should be deleted on an idle system only. See the section “Eliminating Virtual Terminal Lines Example” at the end of this chapter for an example of how to eliminate virtual terminal lines.

Configure Virtual Asynchronous Interfaces

You can configure asynchronous protocol features on virtual terminal lines. Asynchronous interfaces are physical interfaces; you create virtual terminal lines with the **line vty** command. When you configure vritual terminal lines for asynchronous protocol features, you are creating a *virtual asynchronous interface*.

SLIP and PPP can normally function only on asynchronous interfaces, and not on vritual terminal lines. However, extending asynchronous functionality to vritual terminal lines permits you to run SLIP and PPP on on these virtual asynchronous interfaces. Refer to “Enable SLIP and PPP on Virtual Asynchronous Interfaces” in the “Configuring SLIP and PPP” chapter for more information about configuring virtual asynchronous interfaces.

Define Terminal Operation Characteristics

In line configuration mode, you can set terminal operation characteristics that will be in operation for that line until the next time you change the line parameters.

Alternatively, you can temporarily change the line settings with the **terminal** EXEC commands described in the *Cisco Access Connection Guide*.

You can define terminal operation characteristics, as described in the following sections:

- Specify the Terminal Type
- Set the Terminal Screen Length and Width
- Establish Terminal Session Limits
- Define Escape Character and Other Key Sequences
- Specify the International Character Display
- Set the Device Communication Parameters
- Configure Automatic Baud Rate Detection
- Set Character Padding

Specify the Terminal Type

You can specify the type of terminal connected to a line. This feature has two benefits: it provides a record of the type of terminal attached to a line, and it can be used in Telnet terminal negotiations to inform the remote host of the terminal type for display management. To specify the terminal type, perform the following task in line configuration mode:

Task	Command
Specify the terminal type.	**terminal-type** *terminal-name*

Set the Terminal Screen Length and Width

By default, the communication server provides a screen display of 24 lines by 80 characters. You can reconfigure these values if they do not meet the needs of your terminal. Perform the following tasks in line configuration mode:

Task	Command
Set the screen length.	**length** *screen-length*
Set the screen width.	**width** *characters*

The values set can be learned by some host systems that use this type of information in terminal negotiation. Set a value of zero for the screen length to disable pausing between screens of output.

Establish Terminal Session Limits

You might need to control terminal sessions in high traffic areas to provide resources for all users. You can define these limitations for terminal sessions:

- The maximum number of sessions
- The idle session timeout interval or the absolute timeout interval

Task	Command
Set the maximum number of sessions.	**session-limit** *session-number*
Set the idle session timeout interval. or Set the absolute timeout interval.	**session-timeout** *minutes* [**output**] or **absolute-timeout** *minutes*
Warn users of impending timeouts set with the absolute-timeout command.	**logout-warning** [*seconds*]

Note The **absolute-timeout** command overrides any timeouts set through ARAP.

Define Escape Character and Other Key Sequences

You can define or modify the default key sequences to execute functions for system escape, terminal activation, disconnect, and terminal pause. To define or change the default sequence, perform one or more of the following tasks in line configuration mode:

Task	Command
Change the system escape sequence. The escape sequence indicates that the codes that follow have special meaning. The default sequence is Ctrl-^.	**escape-character** *ascii-number*
Define a session activation sequence or character. Typing this sequence at a vacant terminal begins a terminal session. The default key is Return.	**activation-character** *ascii-number*
Define the session disconnect sequence or character. Typing this sequence at a terminal ends the session with the communication server. There is no default sequence.	**disconnect-character** *ascii-number*
Define the hold sequence or character that causes output to the terminal screen to pause. 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.	**hold-character** *ascii-number*

You can reinstate the default value for the escape character or activation character by using the **no** form of the command. For example, issuing the **no escape-character** line configuration command returns the escape character to Ctrl-^.

Note If you are using the autoselect function, the activation character should not be changed from the default value of Return. If you change this default, the autoselect feature may not function immediately.

Specify the International Character Display

You can use a 7-bit character set (such as ASCII) or you can enable a full 8-bit international character set (such as ISO 8859) to allow special graphical and international characters for use in banners and prompts, and to add special characters such as software flow control. These settings can be configured globally, by interface, and locally at the user level. Use the following criteria for determining which configuration mode to use to set up this feature:

- If a large number of connected terminals support nondefault ASCII bit settings, use the global configuration commands.
- If only a few of the connected terminals support nondefault ASCII bit settings, use line configuration commands or the EXEC local terminal setting commands.

Note Setting the EXEC character width to eight bits can cause failures. If a user on a terminal that is sending parity enters the command **help**, an “unrecognized command” message appears because the system is reading all eight bits, although the eighth bit is not needed for the **help** command.

To specify a character set on a global basis, perform one or both of the following tasks in global configuration mode:

Task	Command
Specify the character set used in EXEC and configuration command characters.	**default-value exec-character-bits** {**7** | **8**}
Specify the character set used in special characters such as software flow control, hold, escape, and disconnect characters.	**default-value special-character-bits** {**7** | **8**}

To specify a character set based on hardware, software, or on a per-line basis, perform the appropriate task in line configuration mode, as follows:

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

Note If you are using the autoselect function, the activation character should be set to the default Return, and exec-character-bits to 7. If you change these defaults, the application will not recognize the activation request.

Set the Device Communication Parameters

The communication server 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 line configuration mode:

Task	Command
Set the line speed. Choose from line speed, transmit speed, or receive speed.	**speed** *bps* **txspeed** *bps* **rxspeed** *bps*
Set the data bits.	**databits** {**5** | **6** | **7** | **8**}
Set the stop bits.	**stopbits** {**1** | **1.5** | **2**}
Set the parity bit.	**parity** {**none** | **even** | **odd** | **space** | **mark**}

Configure Automatic Baud Rate Detection

You can configure a terminal to automatically detect the baud rate being used over an asynchronous serial line. To set up automatic baud detection, perform the following task in line configuration mode:

Task	Command
Set the terminal to automatically detect the baud rate.	**autobaud**

To start communications using automatic baud detection, type multiple Returns at the terminal. A 600-, 1800-, or 19200-baud line requires three Returns to detect the baud rate. A line at any other baud rate requires only two Returns. If you type extra Returns after the baud rate is detected, the EXEC simply displays another system prompt.

Set Character Padding

You can change the character padding on a specific output character. 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. To set character padding, perform the following task in line configuration mode:

Task	Command
Set padding on a specific output character for the specified line.	**padding** *ascii-number count*

Record the Device Location

You may record the location of a serial device. The text provided for the location appears in the output of the EXEC monitoring commands.

Task	Command
Record the location of a serial device.	**location** *text*

Configure Software Flow Control

You can set the following types of flow control between the communication server and devices attached to it:

- Software flow control, both in the incoming and outgoing direction
- Hardware flow control

Task	Command
Set the terminal flow control; select software or hardware flow control.	**flowcontrol** {**none** | **software** [**in** | **out**] | **hardware** [**in** | **out**]}

Configure Software Flow Control Start and End Characters

You can define characters or character sequences that signal the start and end of data transmission when software flow control is in effect. This is useful for providing control of data over the serial line.

Task	Command
Set the flow control start character.	**start-character** *ASCII-number*
Set the flow control stop character.	**stop-character** *ASCII-number*

Define a Command String for Automatic Execution

You can set up a command or string of commands that will automatically execute upon connection to another host. Any appropriate EXEC command and any switch or host name that occurs with the EXEC command is allowed. To define a command string, perform the following task in line configuration mode:

Task	Command
Define a command or string of commands to be automatically executed.	**autocommand** *command*

Configure Automatic Protocol Startup

To configure the communication server to allow an AppleTalk Remote Access (ARA), PPP, or SLIP session to start automatically, perform the following task in line configuration mode:

Task	**Command**
Configure a line to automatically start an ARA, PPP or SLIP session.	**autoselect** {**arap | ppp | slip**} | **during login**

The **autoselect** command permits the communication server to allow an appropriate process to start automatically when a starting character is received. The communication server detects either a Return character, which is the start character for an EXEC session, or the start character for the ARA protocol. By using the optional **during login** argument, the username or password prompt is displayed without pressing the Return key. While the Username or Password name is presented, you can choose to answer these prompts or to start sending packets from an autoselected protocol. Refer to the end of this chapter for configuration examples.

Note When using autoselect, the activation character should be set to the default Return, and exec-character-bits to 7. If you change these defaults, the application will not recognize the activation request.

Create Character and Packet Dispatch Sequences

The communication server supports configuration of dispatch sequences and TCP state machines that transmit packets of data upon receipt of the defined character or sequence of characters. You can set up dispatch characters that allow packets to be buffered, then transmitted upon receipt of a character. You can set up a state machine that allows packets to be buffered then transmitted upon receipt of a sequence of characters. This allows for packet transmission by pressing a function key, which is typically defined as a sequence of characters ("Esc I C," for example).

State machines allow control of TCP processes based upon a set of inputs. The current state of the device determines what will happen next given an expected input. The state-machine commands configure the server to search for and recognize a particular sequence of characters, then cycle through a set of states. The user defines these states and up to eight states can be defined. (Think of each state as a step the server takes based upon the assigned configuration commands, and the type of information received.)

The software supports user-specified state machines for determining whether data from an asynchronous port should be sent to the network. This is an extension of the concept of the dispatch character and allows, for example, the equivalent of multicharacter dispatch strings.

Up to eight states can be set up for the state machine. Data packets are buffered until the appropriate character or sequence triggers the transmission. Delay and timer metrics allow for more efficient use of system resources. Characters defined in the TCP state machine take precedence over those defined for a dispatch character.

Perform the following tasks, as needed, for your particular system needs.

Task	Command
Specify the transition criteria for the states in a TCP state machine.	**state-machine** *name state firstchar lastchar* [*nextstate* | **transmit**]
Specify the state machine for TCP packet dispatch.	**dispatch-machine** *name*
Define a character that triggers packet transmission.	**dispatch-character** *ASCII-number* [*ASCII-number2 . . . ASCII-number*]
Set the dispatch timer.	**dispatch-timeout** *milliseconds*

Enable the Auxiliary Port

A backup RS-232 DTE port is available on the ASM-CS and Cisco 2500.

Note This section applies to the ASM-CS and Cisco 2500 only. There is no auxiliary port on the 500-CS.

Use this port to attach to an RS-232 port of a CSU/DSU, protocol analyzer, or modem. The ports assert DTR only when a Telnet connection is established. This auxiliary port does not use RTS/CTS handshaking for flow control.

Task	Command
Enable the auxiliary serial RS-232 DTE port available on the ASM-CS.	**line aux** *line-number*

You cannot use the auxiliary port as a second console port, nor can you initiate connections from this port. Its purpose is to *receive* connections from remote systems. You must order a special cable from your technical support personnel for use with this auxiliary port.

Specify the Transport Protocol for a Specific Line

Use the **transport preferred** command to specify a preferred protocol to be used on connections. Use the **transport input** and **transport output** commands to explicitly specify the protocols allowed on individual lines for both incoming and outgoing connections.

For access servers that support LAT, the default protocol for outgoing connections is LAT. For those that do not support LAT, the default protocol for outgoing connections is Telnet. For incoming connections, the default protocol is **all**.

Perform one or more of the following tasks in line configuration mode to specify transport protocols:

Task	Command
Define which protocols can be used to connect to a specific line of the access server.	**transport input** {**lat** | **mop** | **none** | **pad** | **rlogin** | **telnet** | **all**}
Determine the protocols that can be used for outgoing connections from a line.	**transport output** {**lat** | **none** | **pad** | **rlogin** | **telnet** | **all**}
Specify the protocol for the access server to use if the user did not specify a protocol.	**transport preferred** {**lat** | **none** | **pad** | **rlogin** | **telnet**}
Prevent errant connection attempts.	**transport preferred none**

The system accepts a host name entry at the EXEC system prompt as a Telnet command. If you mistype the host name, the system 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 mistype a command at the EXEC prompt, the system will not attempt to make a Telnet connection.

Set a Line as Insecure

You can set up a line to appear as an insecure, dial-up line. The information is used by the LAT software, which reports such connections as dial-ups to remote systems.

Task	Command
Set the line as a dial-up line.	**insecure**

In the previous versions of Cisco software, any line that used modem control was reported as dial-up through the LAT protocol; this feature allows more direct control.

Save Local Settings between Sessions

You can configure the communication server to save local parameters set with EXEC **terminal** commands between sessions. This ensures that the parameters the user sets will remain in effect between terminal sessions. This behavior is useful for servers in private offices. By default, user-set parameters are cleared when the session ends.

Task	Command
Save local settings between sessions.	**private**

Set Pending Output Notification

You can set up a line to inform a user who has multiple, concurrent Telnet connections when output is pending on a connection other than the current one.

Task	Command
Set up a line to notify a user of pending output.	**notify**

This task performs the same function on a line as does the local EXEC command **terminal notify**. See the *Access and Communication Servers Command Reference* publication for more information about this command.

Establish and Control the EXEC Process

By default, the communication server starts an EXEC process on all lines; however, you can control EXEC processes, as follows:

- Turn the EXEC on or off.

  A serial printer, for example, should not have an EXEC started.

- Set the idle terminal time-out interval.

  The EXEC command interpreter waits for a specified interval of time until the user starts input. If no input is detected, the EXEC resumes the current connection, or if no connections exist, it returns the terminal to the idle state and disconnects the incoming session. To control the EXEC process, perform the following tasks in line configuration mode:

Task	Command
Turn on EXEC processes.	**exec**
Set the idle terminal timeout interval.	**exec-timeout** *minutes* [*seconds*]

Configure Modem Lines

Cisco Systems communication servers use six EIA/TIA-232 signals for each port, an arrangement that allows eight connections to be handled by one 50-pin Telco, RJ-11, or RJ-45 connector. The communication server can support the most popular forms of modem control and hardware flow control as well as high-speed dial-up modems.

The EIA/TIA-232 output signals are Transmit Data (TXDATA), Data Terminal Ready (DTR), and Ready To Send (RTS, 2500 only). The input signals are Receive Data (RXDATA), Clear to Send (CTS), and RING. The sixth signal is ground. Depending on the type of modem control, these names may or may not correspond to the standard EIA/TIA-232 signals, which have similar names.

Dial-up modems that operate over normal dial-up telephone lines at speeds of 9600 bits per second and higher are now available. These modems do not operate at a guaranteed throughput; instead, they operate at a speed dependent on the quality of the line, the effectiveness of data compression algorithms on the data being transmitted, and other variables. These modems use hardware flow control to stop the data from the host by toggling an EIA/TIA-232 signal when their limit is reached.

In addition to hardware flow control, dial-up modems require special software handling. For example, they must be configured to create an EXEC when a user dials in and to hang up when the user exits the EXEC. These modems must also be configured to close any existing network connections if the telephone line hangs up in the middle of a session.

Your communication server supports hardware flow control on its CTS input, which is also used by the normal modem handshake.

You can configure the following modem line characteristics and modem features on the communication server:

- Configure Automatic Dialing
- Close Modem Connections
- Automatically Answer a Modem
- Support a Dial-In Modem
- Support Reverse Modem Connections and Prevent Incoming Calls
- Support Dial-In and Dial-Out Modems
- Configure a Line Timeout Interval
- Configure Automatic Line Disconnect

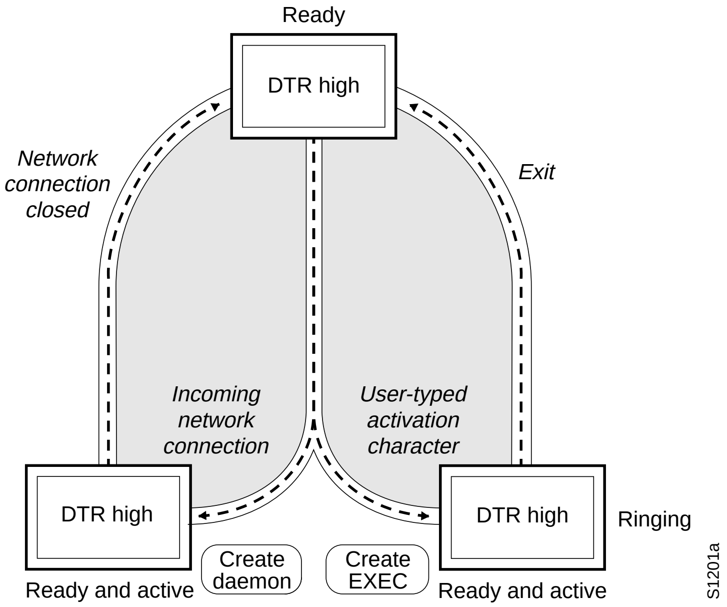
State diagrams accompany some of the tasks in the following sections to illustrate how the modem control works. The diagrams show two processes:

- The “create daemon” process creates a TTY daemon that handles the incoming network connection.
- The “create EXEC” process creates the process that interprets user commands. (Refer to Figures 4-1 through 4-6.)

In the diagrams, the current signal state and the signal the line is watching are listed inside each box. The state of the line (as displayed by the EXEC command **show line**) is listed next to the box. Events that change that state appear in italics along the event path, with actions that the software takes described within the ovals.

Figure 4-1 illustrates line behavior when no modem control is set. The DTR output is always high, and CTS and RING are completely ignored. The communication server creates an EXEC when the user types the activation character. Incoming TCP connections occur instantly if the line is not in use and can be closed only by the remote host.

Figure 4-1 EXEC and Daemon Creation on a Line with No Modem Control



Configure Automatic Dialing

With the dial-up capability, you can set a modem to automatically dial the phone number of a remote communication server. This feature offers cost savings because phone line connections are made as needed. You only pay for using the phone line when there is data to be received or sent. To configure a line for automatic dialing, perform the following task in line configuration mode:

Task	Command
Configure a line to initiate automatic dialing.	**modem dtr-active**

Close Modem Connections

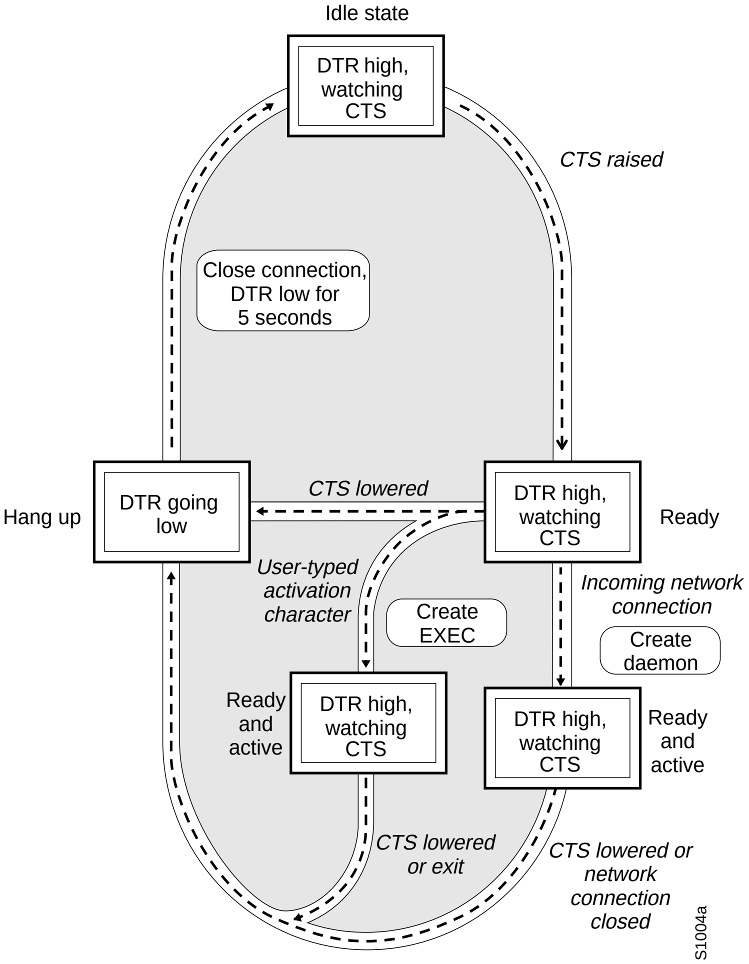
You can configure a line to close connections from a user's terminal when the terminal is turned off, and prevent inbound connections to devices that are out of service. To do so, perform the following task in line configuration mode:

Task	Command
Configure a line to close connections.	**modem cts-required**

Figure 4-2 illustrates the **modem cts-required** process in the context of a continuous CTS. This form of modem control requires that CTS be high throughout the use of the line. If CTS is not high, the user's typed input is ignored and incoming connections are refused (or step to the next line in a rotary group).

Note In order for a communication server to reliably detect a CTS signal change, the signal must remain in the new state for at least one full second.

Figure 4-2 EXEC and Daemon Creation on a Line Configured for Continuous CTS



Automatically Answer a Modem

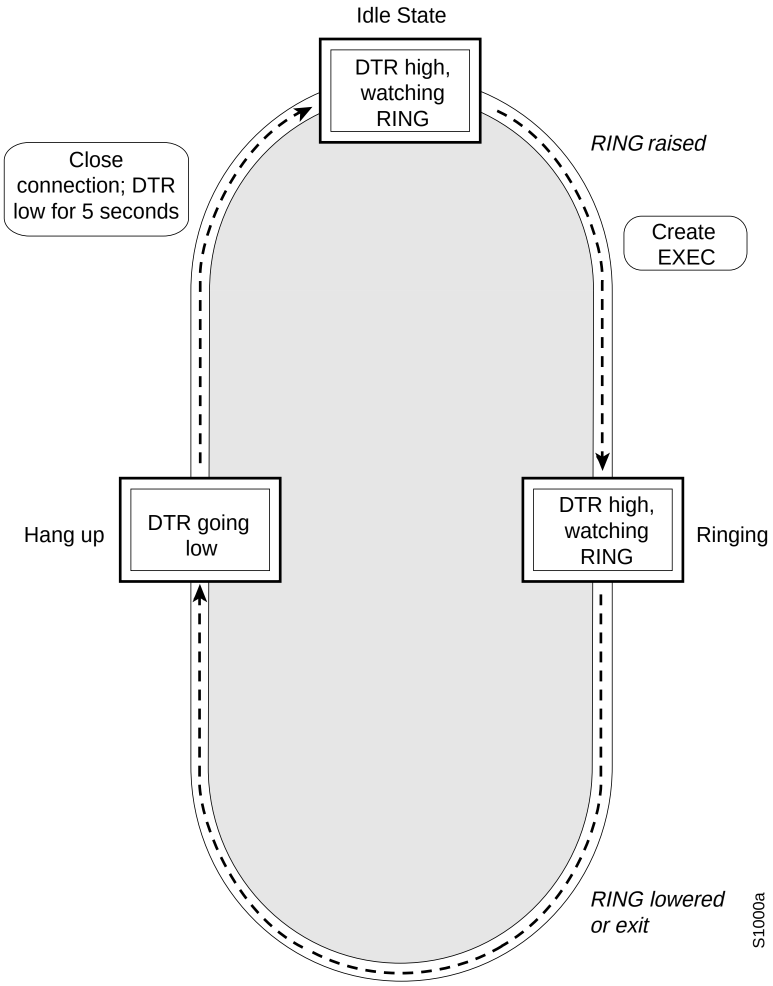
You can configure a line to automatically answer a modem. You would also configure the modem to answer the telephone on its own as long as DTR is high, drop connections when DTR is low, and use its Carrier Detect (CD) signal to accurately reflect the presence of carrier (configuring the modem is a modem-dependent process). Wire the modem's CD signal (generally pin-8) to the communication server's RING input pin-22), and perform the following task in line configuration mode:

Task	Command
Configure a line to automatically answer a modem.	**modem ri-is-cd**

You can turn on the modem's hardware flow control independently to act on the status of the communication server's CTS input. Wire CTS to whatever signal the modem uses for hardware flow control. If the modem expects to control hardware flow in both directions, you might also need to wire the modem's flow control input to some other signal that the communication server always has high (such as DTR).

Figure 4-3 illustrates the **modem ri-is-cd** process with a high-speed dial-up modem. When the communication server detects a signal on the RING input of an idle line, it starts an EXEC or autobaud process on that line. If the RING signal disappears on an active line, the communication server closes any open network connections and terminates the EXEC. If the user exits the EXEC or the communication server terminates it because of no user input, the line hangs up the modem by lowering the DTR signal for five seconds. After five seconds, the modem is ready to accept another call.

Figure 4-3 EXEC Creation on a Line Configured for a High-Speed Dial-up Modem



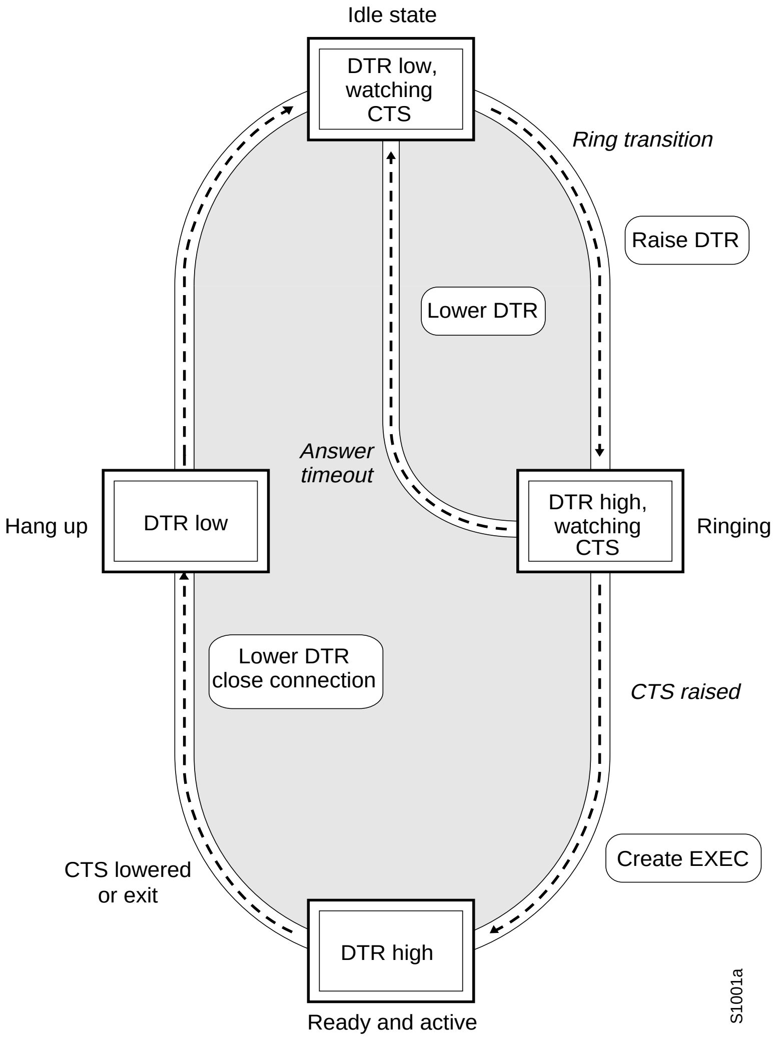
Support a Dial-In Modem

The communication server supports dial-in modems that use DTR to control the off-hook status of the telephone line. Perform the following task in line configuration mode to configure the line to support this feature:

Task	Command
Configure a line for a dial-in modem.	**modem callin**

Figure 4-4 illustrates the **modem callin** process. When a modem dialing line is idle, it has DTR in a low state and waits for a transition to occur on the RING input. This transition causes the line to raise DTR and start watching the CTS signal from the modem. After the modem raises CTS, the communication server creates an EXEC on the line. If the timeout interval (set with the **modem answer-timeout** command) passes before the modem raises CTS, the line lowers DTR and returns to the idle state.

Figure 4-4 EXEC Creation on a Line Configured for Modem Callin



Note The **modem callin** and **modem cts-required** line configuration commands are useful for SLIP operation. These commands ensure that when the line is hung up or CTS drops, the line reverts from SLIP mode to normal interactive mode. These commands do not work if you put the line in network mode permanently.

Although you can use the **modem callin** line configuration command with newer modems, the **modem ri-is-cd** line configuration command described earlier in this section is more appropriate. The **modem ri-is-cd** command frees up CTS for hardware flow control. Modern modems do not require the assertion of DTR to take a phone line off-hook.

Support Reverse Modem Connections and Prevent Incoming Calls

In addition to initiating connections, the communication server can receive incoming connections. This capability allows you to attach serial and parallel printers, modems, and other shared peripherals to the communication server and drive them remotely from other systems. The communication server supports reverse TCP, XRemote, and LAT connections.

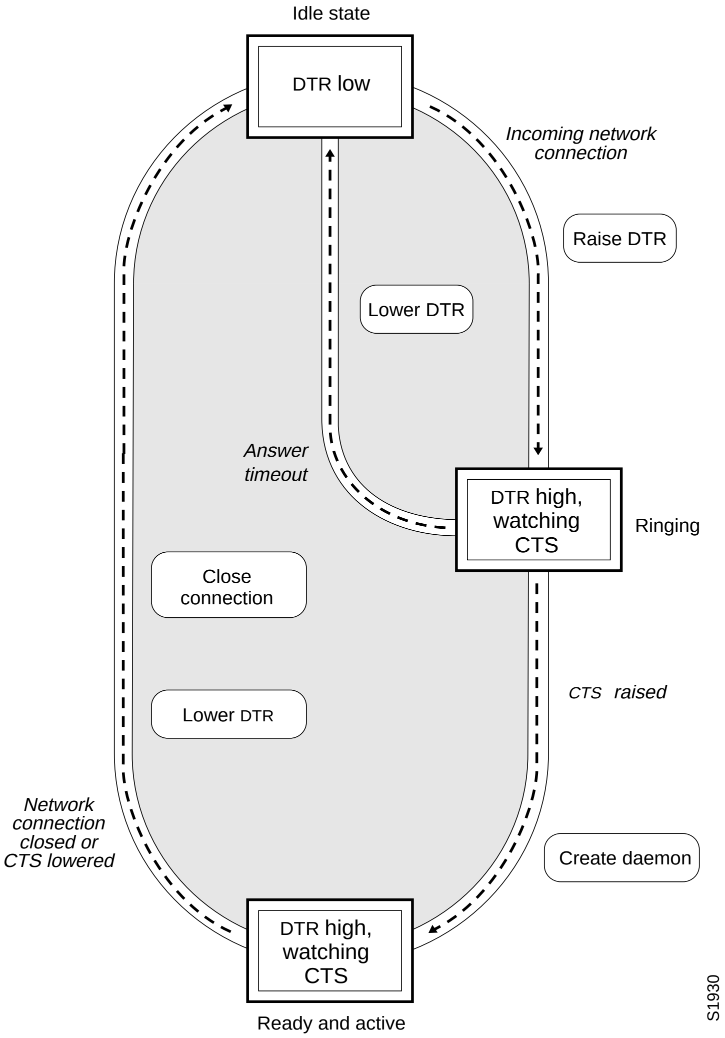
The specific TCP port or socket to which you attach the device determines the type of service the communication server provides on that line. When you attach the serial lines of a computer system or a data terminal switch to the serial lines of the communication server, the communication server acts as a network front-end for a host that does not support the TCP/IP protocols. This arrangement is sometimes called *front-ending*, or *reverse connection mode*.

The communication server supports ports connected to computers that are to be connected to modems. You can configure the communication server to behave somewhat like a modem by performing the following task in line configuration mode. This command also prevents incoming calls.

Task	Command
Configure a line for reverse connections and prevent incoming calls.	**modem callout**

Figure 4-5 illustrates the **modem callout** process. When the communication server receives an incoming connection, it raises DTR and waits to see if the CTS becomes high as an indication that the host has noticed its signal. If the host does not respond within the interval set with the **modem answer-timeout** subcommand, the communication server lowers DTR and drops the connection.

Figure 4-5 Daemon Creation on a Line Configured for Modem Callout



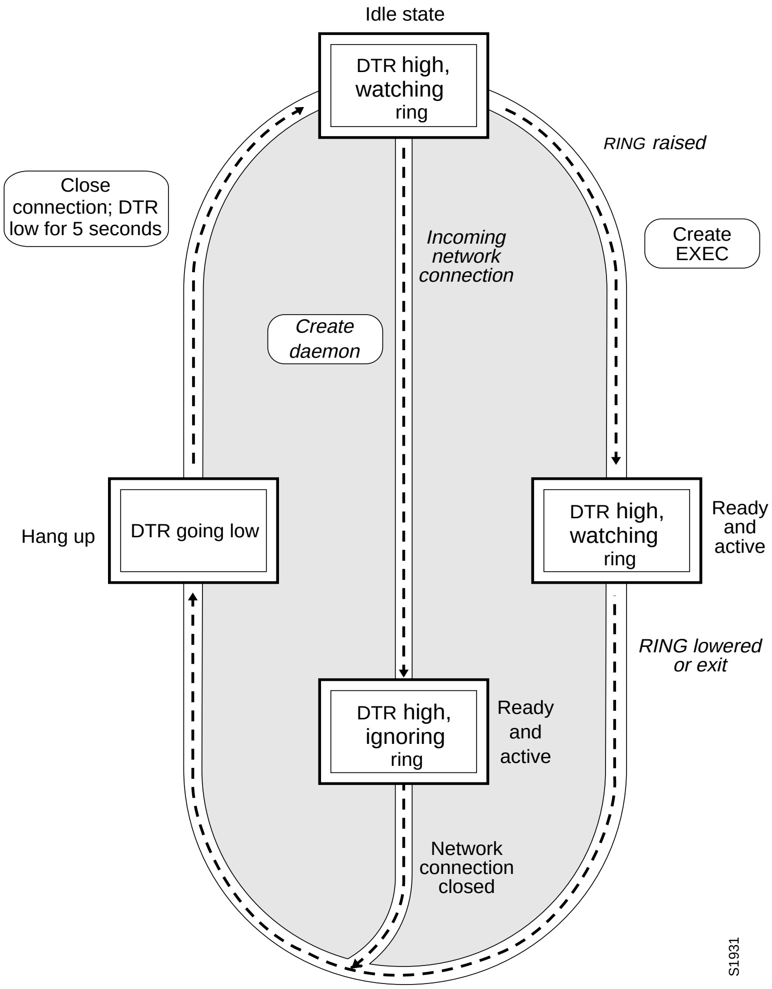
Support Dial-In and Dial-Out Modems

You can configure a line for both incoming and outgoing calls by performing the following task in line configuration mode:

Task	Command
Configure a line for both incoming and outgoing calls.	**modem in-out**

Figure 4-6 illustrates the **modem in-out** process. If the line is activated by raising RING, it behaves exactly as a line configured with the **modem ri-is-cd** line configuration command described earlier. If the line is activated by an incoming TCP connection, the line behaves similarly to a nonmodem line.

Figure 4-6 EXEC and Daemon Creation on a Line Configured for Incoming and Outgoing Calls



Note If your system incorporates dial-out modems, consider using access lists to prevent unauthorized use.

Configure a Line Timeout Interval

You can change the interval that the communication server waits for CTS after raising DTR in response to RING from the default of 15 seconds. To do so, perform the following task in line configuration mode. The timeout applies to the **modem callin** command only.

Task	Command
Configure modem line timing.	**modem answer-timeout** *seconds*

Configure Automatic Line Disconnect

You can configure automatic line disconnect by performing the following task in line configuration mode:

Task	Command
Configure automatic line disconnect.	**autohangup**

The command causes the EXEC to issue the **exit** command when the last connection closes. This feature is useful for UNIX-to-UNIX copy program (UUCP) applications that require this behavior, because UUCP scripts cannot issue the command that hangs up the telephone.

Configure High-Speed Modem Support

Dial-up modems that operate over normal dial-up telephone lines at speeds of 9600 bits per second and higher are now available. These modems do not operate at a guaranteed throughput; instead, they operate at a speed dependent on the quality of the line, the effectiveness of data compression algorithms on the data being transmitted, and other variables. These modems use hardware flow control to stop the data from reaching the host by toggling an RS-232 signal when they cannot accept any more.

In addition to hardware flow control, dial-up modems require special software handling. For example, they must be configured to create an EXEC when a user dials in and to hang up when the user exits the EXEC. These modems must also be configured to close any existing network connections if the telephone line hangs up in the middle of a session.

Your communication server supports hardware flow control on its CTS input, which is also used by the normal modem handshake.

Perform the following tasks to configure and use a high-speed modem:

Task		Command
Step 1	In line configuration mode, enable outgoing hardware flow control based on the CTS input.	**flowcontrol hardware**
Step 2	In EXEC mode, display informational messages about modem control events, such as signal transitions and autobaud progress, on the console terminal.	**debug modem** (See Chapter 5 and the *Debug Command Reference*.)
Step 3	In EXEC mode, display the status of a line. In the detailed command output, a Status line with “Idle” identifies inactive **modem ri-is-cd** lines and all other modem lines; a Status line with “Ready” identifies lines in use.	**show line**[1]
Step 4	In EXEC mode, close all the connections on a line and hang up the modem.	**clear line**

[1] These commands are documented in the “User Interface Commands” chapter in the *Access and Communication Servers Command Reference* publication.

Connections to an Individual Line

Connections to an individual line are most useful when a dial-out modem, parallel printer, or serial printer is attached to that communication server line. To connect to an individual line, the remote host or terminal must specify a particular TCP port on the communication server. If Telnet protocols are required, that port is 2000 (decimal) plus the decimal value of the line number. If reverse XRemote is required, that port is 9000 (decimal) plus the decimal value of the line number.

Note Line numbers are octal on the ASM-CS communication servers and decimal on the 500-CS and Cisco 2500. On the ASM-CS, for example, line 40 corresponds to decimal 32; the corresponding TCP port is 2032. On the 500-CS, the corresponding TCP port for line 9 would be 2009. There is no octal-to-decimal translation. The **service decimal-tty** global configuration command controls the specification of octal or decimal line numbers; see the chapter "System Management Commands," in the *Access and Communication Servers Command Reference* for more information about this command.

If a raw TCP stream is required, the port is 4000 (decimal) plus the decimal line number. The raw TCP stream is usually the required mode for sending data to a printer.

The Telnet protocol requires that carriage return characters be translated into carriage returns and line-feed character pairs. You can turn this translation off by specifying the Telnet binary mode option. To specify this option, connect to port 6000 (decimal) plus the decimal line number.

Assume that a laser printer is attached to line 10 of an ASM-CS communication server. Such a printer usually uses XON/XOFF software flow control. Because the communication server will not receive an incoming connection if the line already has a process, you must ensure that an EXEC is not accidentally started. You must therefore configure it as follows:

```
line 10
flowcontrol software
no exec
```

A host that wants to send data to the printer would connect to the communication server on TCP port 4008, send the data, and then close the connection. (Remember that line number 10 octal equals 8 decimal.)

Change the Retry Interval for a Terminal Port Queue

If you attempt to connect to a remote device, such as a printer, and the device is busy, the connection attempt is placed in a terminal port queue. If the retry interval is set too high, and several communication servers (or other devices) are connected to the remote device, your connection connection attempt can have large delays. To change the retry interval for a terminal port queue, perform the following task in global configuration mode:

Task	Command
Change the retry interval for a terminal port queue.	**terminal-queue entry-retry-interval** *interval*

LPD Protocol Support

Cisco supports a subset of the Berkeley UNIX Line Printer Daemon (LPD) protocol used to send print jobs between UNIX systems. This subset of the LPD protocol permits:

- Improved status information
- Cancellation of print jobs
- Confirmation of successful printing and automatic retry for common failure cases
- Use of standard UNIX software

The Cisco implementation of LPD permits you to configure a printer to allow several types of data to be sent as print jobs (for example, PostScript or raw text).

Configure a Printer

To configure a printer for the LPD protocol, perform the following task in global configuration mode:

Task	Command
Configure printer and specify a tty line (or lines) for the device.	**printer** *printername* {**line** *number* | **rotary** *number*} **[newline-convert]**

If you use the **printer** command, you will also have to modify the */etc/printcap* file on the UNIX system to include the definition of the remote printer on the access server. Use the optional **newline-convert** keyword on UNIX systems that do not handle single character line terminators to convert a newline to a carriage-return, linefeed sequence.

The following example includes the configuration of the printer Saturn on the host Memphis:

```
commlpt|Printer on cisco AccessServer:\
    :rm+memphis:rp+saturn:\
    :sd+/usr/spool/lpd/comm1pt:\
    :lf=?var/log/lpd/commlpt:
```

The content of the actual file may differ depending on the configuration of your system.

Configure Rotary Groups

Connections can be made to the next free line in a group of lines, also called a rotary or hunt group. A line can be in only one rotary group; a rotary group can consist of a single line or several contiguous lines. The console line (line 0) cannot be in a rotary group.

To configure a rotary group, perform the following task in line configuration mode:

Task	Command
Add a line to the specified rotary group.	**rotary** *group*

Enable Password Checking at Login

You can enable password checking on a particular line so that the user is prompted to enter a password at the system login screen. You must then also specify a password. Perform the following tasks in line configuration mode:

Task		Command
Step 1	Enable password checking on a per-line basis using the password specified with the **password** command.	**login**
Step 2	Assign a password to a particular line.	**password** *password*

You can enable password checking on a per-user basis, in which case authentication is based on the username specified with the **username** global configuration command, as described in the "Managing the System" chapter. To enable this type of password checking, perform one of the following tasks in line configuration mode:

Task	Command
Enable password checking on a per-user basis using the username and password specified with the **username** global configuration command.	**login local**

Task	Command
Select the TACACS-style user ID and password-checking mechanism.	**login tacacs** or **login authentication {default |** *list-name***}**

Use the **login tacacs** command with TACACS and XTACACS. Use the **login authentication** command with AAA/TACACS+.

By default, virtual terminals require passwords. If you do not set a password for a virtual terminal, it will respond to attempted connections by displaying an error message and closing the connection. Use the **no login** command to disable this behavior and allow connections without a password.

For other access control tasks and password restrictions, including the **enable password** global configuration command that restricts access to privileged mode, see the “Managing the System” chapter in this publication.

Support Reverse Telnet Connections

In addition to initiating connections, the communication server can receive incoming connections on asynchronous lines. This capability allows you to attach serial printers, modems, and other shared peripherals to the communication server and drive them remotely from other systems. The communication server supports reverse Telnet connections.

The specific TCP port or socket to which you attach the peripheral device determines the type of service the communication server provides on that line. When you attach the serial lines of a computer system or a data terminal switch to a line of the communication server, the communication server acts as a network front end for a host that does not support the TCP/IP protocols. This arrangement is sometimes called *front-ending* or *reverse connection mode*.

To connect an asynchronous line on a communication server, the remote host or terminal must specify a particular TCP port on the communication server. If Telnet protocols are required, that port is 2000 (decimal) plus the decimal value of the line number.

If a raw TCP stream is required, the port is 4000 (decimal) plus the decimal line number. The raw TCP stream is usually the required mode for sending data to a printer.

The Telnet protocol requires that carriage return characters be translated into carriage return and linefeed character pairs. You can turn this translation off by specifying the Telnet binary mode option. To specify this option, connect to port 6000 (decimal) plus the decimal line number.

Disable Enhanced Editing Mode

To disable enhanced editing mode and revert to the editing mode of previous software releases, perform the following task in line configuration mode:

Task	Command
Disable the enhanced editing features for a particular line.	**no editing**

You might disable enhanced editing if you have prebuilt scripts, for example, that do not interact well when enhanced editing is enabled. You can reenable enhanced editing mode with the **editing** command.

Set a Terminal-Locking Mechanism

You can enable a terminal-locking mechanism that allows a terminal to be temporarily locked by performing the following task in global configuration mode:

Task	Command
Enable a temporary terminal locking mechanism.	**lockable**

After you configure the line as lockable, you must still issue the **lock** EXEC command to lock the keyboard.

Dedicate a Line to a Particular User

By default, user-set terminal parameters are cleared with the EXEC command **exit** or when the interval set with the **exec-timeout** line configuration command has passed. However, you can configure a line so that user-set terminal parameters are saved between sessions by performing the following task in line configuration mode:

Task	Command
Configure a line so that the terminal parameters the user sets remain in effect between terminal sessions.	**private**

Provide Line Connection Information after the Login Prompt

You can provide the host name, line number, and location each time an EXEC is started or an incoming connection is made. The line number banner appears immediately after the EXEC banner or incoming banner. It is useful for tracking problems with modems because it lists the host and line for the modem connection. Modem type information is also included if applicable.

To provide service line number information, perform the following task in global configuration mode:

Task	Command
Provide service line number information after the EXEC or incoming banner.	**service linenumber**

Set a Line to Act as a Pipe for File Transfers

You can set a line to act as a transparent pipe so that programs such as Kermit, XMODEM, or CrossTalk can download a file across a communication server line. To do so, perform the following task in EXEC mode:

Task	Command
Temporarily set the ability of a line to act as a transparent pipe for file transfers.	**terminal download**[1]

[1] This command is documented in the *Cisco Access and Connection Guide*.

Configure Chat Scripts for Asynchronous Lines

Chat scripts are strings of text used to send commands for modem dialing, logging onto remote systems, and for initializing asynchronous devices connected to an asynchronous line (on a router, chat scripts can be configured on the auxiliary port only). A chat script must be defined for dialing out on asynchronous lines. You can also configure chat scripts so they are executed for other specific events on a line, or so they can be executed manually. Each script is defined for a different event. These events can include the following:

- Line activation
- Incoming connection initiation
- Asynchronous dial-on-demand routing (see the chapter “Configuring Dial-on-Demand Routing”)
- Line resets
- Communication server startup

To use a chat script, perform the following steps:

Step 1 Define the chat script in global configuration mode using the **chat-script** command.

Step 2 Configure the line so that a chat script is activated when a specific event occurs (using the **script** line configuration command), or start a chat script manually (using the **start-chat** privileged EXEC command).

Create a Chat Script

To define a chat script, perform the following task in global configuration mode:

Task	Command
Create a script that will place a call on a modem, log on to a remote system, or initialize an asynchronous device on a line.	**chat-script** *script-name expect send...* [1]

[1] This command is described in the “Dial-on-Demand Routing Commands” chapter in the *Access and Communication Servers Command Reference* publication.

A limited list of keywords are supported along with expect/send pairs. Send strings can have special escape modifiers.

For an example of how to write scripts, refer to “DDR Configuration Examples” in the chapter “Configuring Dial-on-Demand Routing.”

It is recommended that one chat script (a “modem” chat script) be written for placing a call and another chat script (a “system” or “login” chat script) be written to log onto remote systems, where required.

Suggested Chat Script Naming Conventions (for Dial Scripts only)

When you create a script name, you should include the modem vendor, type, and modulation separated by hyphens. For example, if you have a Telebit t3000 modem that uses V.32bis modulation, your script name would be *telebit-t3000-v32bis.*

A suggested naming convention for chat scripts used for dialing is as follows:

vendor-type-modulation

In other words, the syntax of the **chat-script** command becomes the following:

chat-script *vendor-type-modulation expect send...*

For example, if you have a Telebit t3000 modem that uses V.32bis modulation, you would name your chat script as follows:

```
telebit-t3000-v32bis
```

The chat-script command could become the following:

```
cs(config)# chat-script telebit-t3000-v32bis ABORT ERROR ABORT BUSY ABORT
"NO ANSWER" "" "ATH" OK "ATDT\T" TIMEOUT 30 CONNECT
```

Adhering to this naming convention allows you to specify a range of chat scripts using partial chat script names with regular expressions. This is particularly useful for dialer rotary groups and is explained further in the "Configure an Interface to Receive Calls" section in the "Configuring Dial-on-Demand Routing" chapter.

Configure the Line to Activate Chat Scripts

Chat scripts can be activated by any of five events, each corresponding to a different version of the **script** line configuration command. To start a chat script manually at any point, refer to the section "Start a Chat Script Manually on an Asynchronous Line."

To define a chat script to start automatically when a specific event occurs, perform the following tasks in line configuration mode:

Task	Command
Start a chat script on a line when the line is activated (every time a command EXEC is started on the line).	**script activation** *regexp*[1]
*Start a c*hat script on a line when *a network connection is made to the line.*	**script connection** *regexp*
Specify a modem script for DDR on a line.	**script dialer**[2] *regexp*
*Start a c*hat script on a line whenever *the line is reset.*	**script reset** *regexp*
*Start a c*hat script on a line whenever *the system is started up.*	**script startup** *regexp*

[1] The argument regexp is a regular expression that is matched to a script name that has already been defined using the **chat-script** command..

[2] To use a chat script for dial-on-demand routing (DDR), refer to "Create Chat Scripts for Asynchronous Interfaces" in the chapter "Configuring Dial-on-Demand Routing."

Note Outbound chat scripts are not supported on lines where modem control is set for inbound activity only (such as **modem ri-is-cd**).

Start a Chat Script Manually on an Asynchronous Line

You can start a chat script manually on any line that is currently not active by performing the following task in privileged EXEC mode:

Task	Command
Start a chat script manually on any asynchronous line.	**start-chat** *regexp* [*line-number* [*dialer-string*]]

If you do not specify the line number, the script runs on the current line. If the line specified is already in use, you cannot start the chat script. A message appears indicating that the line is already in use.

Display Terminal Banner Messages

You can provide the following types of messages that will be displayed to users of terminals connected to the communication server:

- Configure a Message-of-the-Day (MOTD) Banner
- Configure a Line Activation Message
- Configure an Incoming Message Banner
- Configure an Idle Terminal Message
- Display a “Line in Use” Message
- Display a “Host Failed” Message
- Enable or Disable the Display of Messages

You can also turn on or off message display.

The following sections explain how to configure these messages and how to suppress display of message-of-the-day and line activation banners.

Configure a Message-of-the-Day (MOTD) Banner

You can configure a message-of-the-day (MOTD) to be displayed on all connected terminals. This message is displayed at login and is useful for sending messages that affect all network users, such as impending system shutdowns. To do so, perform the following task in global configuration mode:

Task	Command
Configure a MOTD banner.	**banner motd** *d message d*

Configure a Line Activation Message

You can configure a line activation message to be displayed when an EXEC process such as line activation or incoming connection to a virtual terminal is created. To do so, perform the following task in global configuration mode:

Task	Command
Configure a message to be displayed on terminals with an interactive EXEC.	**banner exec** *d message d*

Configure an Incoming Message Banner

You can configure a message to be displayed on terminals connected to reverse Telnet lines. This message is useful for providing instructions to users of these types of connections. Reverse Telnet connections are described in more detail in the section “Support Reverse Telnet Connections” earlier in this chapter.

To configure the message that will be sent on incoming connections, perform the following task in global configuration mode:

Task	Command
Configure messages to display on terminals connected to reverse Telnet lines.	**banner incoming** *d message d*

Configure an Idle Terminal Message

You can configure messages to be displayed on a console or terminal not in use. Also called a *vacant message*, this message is different from the banner message displayed when an EXEC process is activated. To configure an idle terminal message, perform the following task in line configuration mode:

Task	Command
Display an idle terminal message.	**vacant-message** [*d message d*]

Display a "Line in Use" Message

You can display a "line in use" message when an incoming connection is attempted and all rotary group or other lines are in use. Perform the following task in line configuration mode:

Task	Command
Display a "line-in-use" message.	**refuse-message** *d message d*

If you do not define such a message, the user will receive a system-generated error message when all lines are in use. You can also use this message to provide the user with further instructions.

Display a "Host Failed" Message

You can display a "host failed" message when a Telnet connection with a specific host fails. Perform the following task in line configuration mode:

Task	Command
Display a "host failed" message.	**busy-message** *hostname d message d*

Enable or Disable the Display of Messages

You can control display of the message -of-the-day and line activation banners. By default, these banners are displayed on all lines. To suppress or reinstate the display of such messages, perform one of the following tasks in line configuration mode:

Task	Command
Suppress banner display.	**no exec-banner**
Reinstate the display of the EXEC or MOTD banners.	**exec-banner**

Line Configuration Examples

The following sections provide line configuration examples:

- Basic Line Configuration Example
- Creating Additional Virtual Terminal Lines Example
- Eliminating Virtual Terminal Lines Example
- Banner Message Example
- Password Checking Examples

Basic Line Configuration Example

In the following example, the user configures console line 0, auxiliary line 0, and virtual terminal lines 0 through 4:

```
line vty 0 4
login
line con 0
password poPPee
line aux 0
password Mypassword
no exec
access-class 1 in
speed 19200
line vty 0
exec-timeout 0 0
password Mypassword
line vty 1
exec-timeout 0 0
password Mypassword
line vty 2
exec-timeout 0 0
password Mypassword
line vty 3
password Mypassword
line vty 4
password Mypassword
```

Creating Additional Virtual Terminal Lines Example

In the following example, the user creates and configures the maximum 100 virtual terminal lines with the “no login” feature:

```
line vty 0 99
no login
```

Eliminating Virtual Terminal Lines Example

In the following example, the user eliminates virtual terminal line number 5 and all higher-numbered virtual terminal lines. Only virtual terminal lines 0 to 4 will remain.

```
no line vty 5
```

Banner Message Example

The following example shows how to use the **banner** global configuration command and **no exec-banner** line configuration command to notify your users that the server is going to be reloaded with new software:

```
! Both messages are inappropriate for the VTYs.
line vty 0 4
```

```
no exec-banner
!
banner exec /
This is Cisco Systems training group communication server.

Unauthorized access prohibited.
/
!
banner incoming /
You are connected to a Hayes-compatible modem.

Enter the appropriate AT commands.
Remember to reset anything to change before disconnecting.
/
!
banner motd /
The communication server will go down at 6pm for a software upgrade
/
```

Password Checking Examples

The following example shows password checking enabled for a virtual terminal line 1:

```
line vty 1
login
password letmein
```

The following example shows password checking enabled on a user basis:

```
username jksmith password 0 letmein
username lmjones password 0 littlerock
...
line vty 1
login local
```