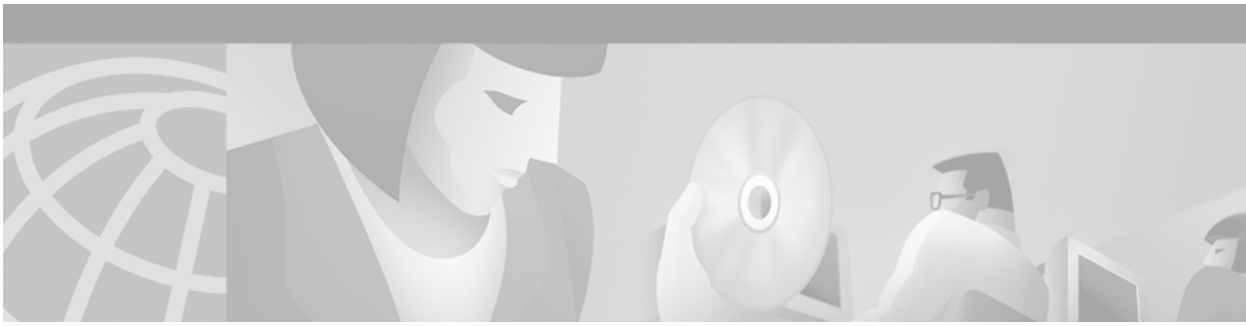

Configuring Operating Characteristics for Terminals

This chapter describes how to configure operating characteristics for terminals. For a complete description of the terminal operation commands in this chapter, refer to the "Terminal Operating Characteristics Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

Terminal Operating Characteristics Configuration Task List

To configure operating characteristics for terminals, perform any of the tasks described in the following sections. All tasks in this chapter are optional.

- Displaying Information About the Current Terminal Session
- Setting Local Terminal Parameters
- Saving Local Settings Between Sessions
- Ending a Session
- Changing Terminal Session Parameters
- Displaying Debug Messages on the Console and Terminals
- Recording the Serial Device Location
- Changing the Retry Interval for a Terminal Port Queue
- Configuring LPD Protocol Support on a Printer

Note For additional information about configuring terminal services, see the Release 12.2 *Cisco IOS Terminal Services Configuration Guide* and the Release 12.2 *Cisco IOS Dial Technologies Configuration Guide*.

Displaying Information About the Current Terminal Session

To display terminal line information, use the following commands in user or privileged EXEC mode, as needed:

Command	Purpose
`Router> show whoami text`	Displays information about the terminal line being used for the current session, including host name, line number, line speed, and location. If text is included as an argument in the command, that text is displayed as part of the additional data about the line.
`Router> where`	Lists all open sessions associated with the current terminal line. An asterisk (*) in the output indicates the current terminal session.

The following example shows sample output of the **show whoami** command:

```
Router> show whoami

Comm Server "Router", Line 0 at 0bps.  Location "Second floor, West"

--More--
Router>
```

To prevent the information from disappearing from the screen, the **show whoami** command always displays a --More-- prompt before returning to the CLI prompt. Press the Spacebar to return to the prompt.

Setting Local Terminal Parameters

The **terminal** EXEC mode commands enable or disable features for the current session only. You can use these commands to temporarily change terminal line settings without changing the stored configuration file.

To display a list of the commands for setting terminal parameters for the current session, use the following command in EXEC mode:

Command	Purpose
`Router# terminal ?`	Lists the commands for setting terminal parameters.

The following example shows sample output for the **terminal ?** command. Commands available on your routing device will vary depending on the software image and hardware you are using.

```
Router> terminal ?
  autohangup              Automatically hangup when last connection closes
  data-character-bits     Size of characters being handled
  databits                Set number of data bits per character
  dispatch-character      Define the dispatch character
  dispatch-timeout        Set the dispatch timer
  download                Put line into 'download' mode
  editing                 Enable command line editing
  escape-character        Change the current line's escape character
  exec-character-bits     Size of characters to the command exec
  flowcontrol             Set the flow control
```

```
full-help               Provide help to unprivileged user
help                    Description of the interactive help system
history                 Enable and control the command history function
hold-character          Define the hold character
ip                      IP options
keymap-type             Specify a keymap entry to use
lat                     DEC Local Area Transport (LAT) protocol-specific
                        configuration
length                  Set number of lines on a screen
no                      Negate a command or set its defaults
notify                  Inform users of output from concurrent sessions
padding                 Set padding for a specified output character
parity                  Set terminal parity
rxspeed                 Set the receive speed
special-character-bits  Size of the escape (and other special) characters
speed                   Set the transmit and receive speeds
start-character         Define the start character
stop-character          Define the stop character
stopbits                Set async line stop bits
telnet                  Telnet protocol-specific configuration
telnet-transparent      Send a CR as a CR followed by a NULL instead of a CR
                        followed by a LF
terminal-type           Set the terminal type
transport               Define transport protocols for line
txspeed                 Set the transmit speeds
width                   Set width of the display terminal
```

Throughout this chapter, many terminal settings can be configured for all terminal sessions or for just the current terminal session. Settings for all terminal sessions are configured in line configuration mode and can be saved. Settings for the current session are specified using EXEC mode commands that generally begin with the word **terminal**.

Saving Local Settings Between Sessions

You can configure the Cisco IOS software to save local parameters (set with **terminal** EXEC mode commands) between sessions. Saving these local settings ensures that the parameters the user sets will remain in effect between terminal sessions. This function is useful for servers in private offices. To save local settings between sessions, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# private`	Saves local settings between sessions.

If the **private** line configuration command is not used, user-set terminal parameters are cleared when the session ends with either the **exit** EXEC mode command or when the interval set with the **exec-timeout** line configuration command has passed.

Ending a Session

To end a session, use the following command in EXEC mode:

Command	Purpose
`Router>` **`quit`**	Ends the current session.

Refer to the "Managing Connections, Menus, and System Banners" chapter for more information on ending sessions and closing connections.

Changing Terminal Session Parameters

This section explains how to change terminal and line settings both for a particular line and locally. The local settings are set with the **terminal** EXEC mode commands. They temporarily override the settings made by the system administrator and remain in effect only until you exit the system. 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.

The following sections describe the tasks used to make the more common changes to the terminal and line settings:

- Defining the Escape Character and Other Key Sequences
- Specifying Telnet Operation Characteristics
- Configuring Data Transparency for File Transfers
- Specifying an International Character Display

The following sections describe the tasks used to make the less common changes to the terminal and line settings:

- Setting Character Padding
- Specifying the Terminal and Keyboard Type
- Changing the Terminal Screen Length and Width
- Enabling Pending Output Notifications
- Creating Character and Packet Dispatch Sequences
- Changing Flow Control for the Current Session
- Enabling Session Locking
- Configuring Automatic Baud Rate Detection
- Setting a Line as Insecure
- Configuring Communication Parameters for Terminal Ports

Defining the Escape Character and Other Key Sequences

You can define or modify the default keys used to execute functions for system escape, terminal activation, disconnect, and terminal pause. Generally, the keys used are actually combinations of keys, such as pressing the Control (Ctrl) key and another key (or keys) at the same time (such as Ctrl-^).

Sequences of keys, such as pressing the Control key and another key, then pressing yet another key, are also sometimes used (for example Ctrl-^, x). However, in each case these keys are referred to as characters, because each key or combination of keys is represented by a single ASCII character. For a complete list of available ASCII characters and their decimal and keyboard equivalents, see the "ASCII Character Set" appendix of the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

Globally Defining Escape Character and Other Key Sequences

To define or change the default key sequences involved with terminal session activation, disconnection, escape, or pausing, use the following commands in line configuration mode, as needed:

Command	Purpose
`Router(config-line)# escape-character {ascii-number | ascii-character | break | default | none}`	Changes the system escape character. We recommend the use of the ASCII characters represented by the decimal numbers 1 through 30. The escape character can be a single character (such as ‘), a key combination (such as Ctrl-X), or a sequence of keys (such as Ctrl-^, X). The default escape character (key combination) is Ctrl-Shift-6 (Ctrl-^), or Ctrl-Shift-6, X (Ctrl-^, X).
`Router(config-line)# activation-character ascii-number`	Defines a session activation character. Entering this character at a vacant terminal begins a terminal session. The default activation character is the Return key.
`Router(config-line)# disconnect-character ascii-number`	Defines the session disconnect character. Entering this character at a terminal ends the session with the router. There is no default disconnect character.
`Router(config-line)# hold-character ascii-number`	Defines the hold character that causes output to the screen to pause. After this character has been set, a user can enter the character at any time to pause output to the terminal screen. To resume output, the user can press any key. To use the hold character in normal communications, precede it with the escape character. There is no default hold character.

For most of the commands described, you can reinstate the default value by using the **no** form. However, to return the escape character to its default, you should use the **escape-character default** line-configuration command.

Note If you are using the autoselect function (enabled using the **autoselect** line configuration command), the activation character should not be changed from the default value of Return. If you change this default, the autoselect feature may not function.

Defining Escape and Pause Characters for the Current Session

For the current terminal session, you can modify key sequences to execute functions for system escape and terminal pause. To modify these sequences, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router>` **`terminal escape-character`** *`ascii-number`*	Changes the system escape sequence for the current session. The escape sequence indicates that the codes that follow have special meaning. The default key combination is Ctrl-Shift-6 (Ctrl-^).
`Router>` **`terminal hold-character`** *`ascii-number`*	Defines 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.

The **terminal escape-character** EXEC 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 the router to EXEC mode when the router is connected to another device.

Specifying Telnet Operation Characteristics

To set Telnet operation characteristics for access servers, perform the tasks described in the following sections:

- Generating a Hardware Break Signal for a Reverse Telnet Connection
- Setting the Line to Refuse Full-Duplex, Remote Echo Connections
- Allowing Transmission Speed Negotiation
- Synchronizing the Break Signal
- Changing the End-of-Line Character

Note The commands in this section apply only to access servers.

Generating a Hardware Break Signal for a Reverse Telnet Connection

To cause the access server 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, use the following command in EXEC mode:

Command	Purpose
`Router>` **`terminal telnet break-on-ip`**	Generates a hardware Break signal on the EIA/TIA-232 line that is associated with a reverse Telnet connection for the current line and session.

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.

Setting 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 or remote echo options on incoming connections, use the following command in EXEC mode:

Command	Purpose
`Router>` **`terminal telnet refuse-negotiations`**	Sets the current line to refuse to negotiate full-duplex for the current session.

Allowing Transmission Speed Negotiation

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

Command	Purpose
`Router>` **`terminal telnet speed`** *`default-speed maximum-speed`*	Allows the Cisco IOS software to negotiate transmission speed for the current line and session.

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.

Synchronizing 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, use the following command in EXEC mode:

Command	Purpose
`Router> terminal telnet sync-on-break`	Causes the Cisco IOS software to send a Telnet Synchronize signal when it receives a Telnet Break signal on the current line and session.

Changing the End-of-Line Character

The end of each line typed at the terminal is ended with a CR+LF (Carriage Return plus Line Feed) signal. The CR+LF signal is sent when a user presses Enter or Return. To cause the current terminal line to send a CR signal as a CR followed by a NULL instead of a CR followed by a line feed (LF), use the following command in EXEC mode:

Command	Purpose
`Router> terminal telnet transparent`	Causes the current terminal line to send a CR signal as a CR followed by a NULL instead of a CR followed by an LF.

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

Configuring Data Transparency for File Transfers

Data transparency enables the Cisco IOS software to pass data on a terminal connection without the data being interpreted as a control character.

During terminal operations, some characters are reserved for special functions. For example, the key combination Ctrl-Shift-6, X (^^x) suspends a session. When transferring files over a terminal connection (using the Xmodem or Kermit protocols, for example), you must suspend the recognition of these special characters to allow a file transfer. This process is called *data transparency.*

You can set a line to act as a transparent pipe so that programs such as Kermit, Xmodem, and CrossTalk can download a file across a terminal line. To temporarily configure a line to act as a transparent pipe for file transfers, use the following command in EXEC mode:

Command	Purpose
`Router> terminal download`	Configures the terminal line to act as a transparent pipe for file transfers.

The terminal download command is equivalent to using all the following commands:

- **terminal telnet transparent**
- **terminal no escape-character**

- **terminal no hold-character**
- **terminal no padding 0**
- **terminal no padding 128**
- **terminal parity none**
- **terminal databits 8**

Specifying an International Character Display

The classic U.S. ASCII character set is limited to 7 bits (128 characters), which adequately represents most displays in the U.S. Most defaults on the modem router work best on a 7-bit path. However, international character sets and special symbol display can require an 8-bit wide path and other handling.

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). This allows special graphical and international characters for use in banners and prompts, and adds special characters such as software flow control. Character settings can be configured globally, per line, or locally at the user level. Use the following criteria for determining which configuration mode to use when you set this international character display:

- 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 an 8-bit character set can cause failures. If a user on a terminal that is sending parity enters the **help** command, an "unrecognized command" message appears because the system is reading all eight bits, although the eighth bit is not needed for **help**.

If you are using the **autoselect** function, the activation character should be set to the default Return, and the EXEC character bit should be set to 7. If you change these defaults, the application does not recognize the activation request.

Specifying the Character Display for All Lines

To specify a character set for all lines (globally), use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# default-value exec-character-bits {7 | 8}`	Specifies the character set used in command characters.
`Router(config)# default-value special-character-bits {7 | 8}`	Specifies the character set used in special characters such as software flow control, hold, escape, and disconnect characters.

Specifying the Character Display for a Line

To specify a character set based on hardware, software, or on a per-line basis, use any of the following commands in line configuration mode:

Command	Purpose
`Router(config-line)# databits {5 | 6 | 7 | 8}`	Sets the number of data bits per character that are generated and interpreted by hardware.
`Router(config-line)# data-character-bits {7 | 8}`	Sets the number of data bits per character that are generated and interpreted by software.
`Router(config-line)# exec-character-bits {7 | 8}`	Specifies the character set used in EXEC and configuration command characters on a per-line basis.
`Router(config-line)# special-character-bits {7 | 8}`	Specifies the character set used in special characters (such as software flow control, hold, escape, and disconnect characters) on a per-line basis.

Specifying the Character Display for the Current Session

To specify a character set based on hardware, software, or on a per-line basis for the current terminal session, use the following commands in EXEC mode:

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

Setting Character Padding

Character padding adds a number of null bytes to the end of a line and can be used to make that line an expected length for conformity. You can change the character padding on a specific output character.

Setting Character Padding for a Line

To set character padding for a line, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# padding` *ascii-number count*	Sets padding on a specific output character for the specified line.

Changing Character Padding for the Current Session

To change character padding on a specific output character for the current session, use the following command in EXEC mode:

Command	Purpose
`Router>` **`terminal padding`** *`ascii-number count`*	Sets padding on a specific output character for the specified line for the current session.

Specifying the Terminal and Keyboard 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.

Specifying the Terminal Type for a Line

To specify the terminal type for a line, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)#` **`terminal-type`** {*`terminal-type`*}	Specifies the terminal type. Any string is accepted for the *terminal-type* argument.

This feature is used by TN3270 terminals to identify the keymap and ttycap passed by the Telnet protocol to the end host.

Specifying the Terminal and Keyboard Type for the Current Session

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

Command	Purpose
`Router>` **`terminal terminal-type`** *`terminal-type`*	Specifies the terminal type for the current session.

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

Command	Purpose
`Router>` **`terminal keymap-type`** *`keymap-name`*	Specifies the keyboard type for the current session.

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 (using the **terminal-type** line configuration command) and provide these names to terminal users.

Changing the Terminal Screen Length and Width

By default, the Cisco IOS software provides a screen display of 24 lines by 80 characters. You can change these values if they do not meet the requirements of your terminal. The screen values you set are passed during rsh and rlogin sessions.

The screen values set can be learned by some host systems that use this type of information in terminal negotiation. To disable pausing between screens of output, set the screen length to 0.

The screen length specified can be learned by remote hosts. For example, the rlogin protocol uses the screen length to set terminal parameters on a remote UNIX host. The width specified also can be learned by remote hosts.

Setting the Terminal Screen Length and Width for a Line

To set the terminal screen length and width for all sessions on a line, use either of the following commands in line configuration mode, as needed:

Command	Purpose
`Router(config-line)# length screen-length`	Sets the screen length.
`Router(config-line)# width characters`	Sets the screen width.

Setting the Terminal Screen Length and Width for the Current Session

To set the number of lines or character columns on the current terminal screen for the current session, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router> terminal length screen-length`	Sets the screen length for the current session.
`Router> terminal width characters`	Sets the screen width for the current session.

Enabling Pending Output Notifications

You can enable the system to inform users when output is pending on a connection other than the active connection. This feature is for situations in which users are likely to have multiple, concurrent telnet connections through the system. For example, the user might want to know when another connection receives mail or a message.

Enabling Pending Output Notifications for a Line

To enable pending output notifications for a line, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# notify`	Enables a line to notify users of pending output on another connection.

Setting Pending Output Notification for the Current Session

To set pending output notification for the current session, use the following command in EXEC mode:

Command	Purpose
`Router> terminal notify`	Sets up a line to notify a user of pending output for the current session.

Creating Character and Packet Dispatch Sequences

The Cisco IOS software supports dispatch sequences and TCP state machines that send data packets only when they receive a defined character or sequence of characters. You can configure dispatch characters that allow packets to be buffered, then sent upon receipt of a character. You can configure a state machine that allows packets to be buffered, then sent upon receipt of a sequence of characters. This feature enables packet transmission when the user presses a function key, which is typically defined as a sequence of characters, such as Esc I C.

TCP state machines can control TCP processes with a set of predefined character sequences. The current state of the device determines what happens next, given an expected character sequence. 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—up to eight states can be defined. (Think of each state as a task that the server performs based on the assigned configuration commands and the type of character sequences received.)

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

Up to eight states can be configured 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.

Setting Character and Packet Dispatch Sequences for a Line

To configure your system, use the following commands in line configuration mode:

Command	Purpose
`Router(config-line)# state-machine name state firstchar lastchar [nextstate | transmit]`	Specifies the transition criteria for the states in a TCP state machine.
`Router(config-line)# dispatch-machine name`	Specifies the state machine for TCP packet dispatch.
`Router(config-line)# dispatch-character ascii-number [ascii-number2 . . . ascii-number]`	Defines a character that triggers packet transmission.
`Router(config-line)# dispatch-timeout milliseconds`	Sets the dispatch timer.
`Router(config-line)# buffer-length length`	Specifies the maximum length of the data stream to be forwarded.

Changing the Packet Dispatch Character for the Current Session

To change the packet dispatch character for the current session, use the following command in EXEC mode:

Command	Purpose
`Router> terminal dispatch-character ascii-number1 [ascii-number2 . . . ascii-number]`	Defines a character that triggers packet transmission for the current session.

Changing Flow Control for the Current Session

To change flow control between the router and attached device for this session, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router> terminal flowcontrol {none | software [in | out] | hardware}`	Sets the terminal flow control for this session.
`Router> terminal start-character ascii-number`[1]	Sets the flow control start character in the current session.
`Router> terminal stop-character ascii-number`[1]	Sets the flow control stop character in the current session.

1. This command is seldom used. Typically, you only need to use the **terminal flowcontrol** command.

Enabling Session Locking

The **lock** EXEC command temporarily locks access to a session, denying access to other users. Session locking must be enabled on the line for the **lock** command to work. To allow session locking by users on a specific line or group of lines, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# lockable`	Enables a temporary terminal-locking mechanism.

Configuring Automatic Baud Rate Detection

You can configure a line to automatically detect the baud rate being used. To set up automatic baud rate detection, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# autobaud`	Configures a line to automatically detect the baud rate.

Note Do not use the **autobaud** command with the **autoselect** command.

To start communications using automatic baud detection, use 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 use extra Returns after the baud rate is detected, the EXEC facility simply displays another system prompt.

Setting a Line as Insecure

You can set up a terminal line to appear as an insecure dialup line. The information is used by the local-area transport (LAT) software, which reports such dialup connections to remote systems.

To set a line as insecure, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# insecure`	Sets the line as a dialup line.

In early releases of Cisco IOS software, any line that used modem control was reported as dialup connection through the LAT protocol; this command allows more direct control of your line.

Configuring Communication Parameters for Terminal Ports

You can change the following parameters as necessary to meet the requirements of the terminal or host to which you are attached. To do so, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router> terminal {speed | txspeed | rxspeed} bps`	Sets the line speed for the current session. Choose from line speed, transmit speed, or receive speed.
`Router> terminal databits {5 | 6 | 7 | 8}`	Sets the data bits for the current session.
`Router> terminal stopbits {1 | 1.5 | 2}`	Sets the stop bits for the current session.
`Router> terminal parity {none | even | odd | space | mark}`	Sets the parity bit for the current session.

Displaying Debug Messages on the Console and Terminals

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

Command	Purpose
`Router# terminal monitor`	Displays **debug** command output and system error messages in EXEC mode on the current terminal.

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

Recording the Serial Device Location

You can record the location of a serial device. The text provided for the location appears in the output of the EXEC monitoring commands. To record the device location, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# location` *text*	Records the location of a serial device.

Changing the Retry Interval for a Terminal Port Queue

If you attempt to connect to a remote device such as a printer that is busy, the connection attempt is placed in a terminal port queue. If the retry interval is set too high, and several routers or other devices are connected to the remote device, your connection attempt can have long delays. To change the retry interval for a terminal port queue, use the following command in global configuration mode:

Command	Purpose
`Router(config)# terminal-queue entry-retry-interval` *interval*	Changes the retry interval for a terminal port queue.

Configuring LPD Protocol Support on a Printer

The Cisco IOS software 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 the following:

- Improved status information
- Cancellation of print jobs
- Confirmation of printing and automatic retry for common print failures
- 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).

To configure a printer for the LPD protocol, use the following command in global configuration mode:

Command	Purpose
`Router(config)# printer` *printername* {`line` *number* | `rotary` *number*} [`newline-convert`]	Configures a printer and specifies a tty line (or lines) for the device.

If you use the **printer** command, you also must modify the /etc/printcap file on the UNIX system to include the definition of the remote printer on the router. Use the optional **newline-convert** keyword on UNIX systems that do not handle single character line terminators to convert a new line to a character Return, line-feed sequence.

The following example includes the configuration of the printer named 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 UNIX system.

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, use the following command in EXEC mode:

Command	Purpose
`Router> show printer`	Lists currently defined printers and their current usage statistics.

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 the /etc/printcap file on your UNIX system to include the definition of the remote printer in the Cisco IOS software.