

# 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 *Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To configure operating characteristics for terminals, perform any of the tasks in the following sections:

- Display Information about Current Terminal Session
- Set Local Terminal Parameters
- Save Local Settings between Sessions
- End a Session
- Change Terminal Session Parameters
- Record the Device Location
- Change the Retry Interval for a Terminal Port Queue
- LPD Protocol Support

## Display Information about Current Terminal Session

The **show whoami** command displays information about the current user’s terminal line, including hostname, line number, line speed, and location. To display line information, use the following command at the EXEC prompt:

Command	Purpose
<b>show whoami</b> <i>text</i>	Display line information.

If text is included as an argument in the command, that text is displayed as part of the additional data about the line.

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 being lost, this command always displays a More prompt before returning. Press the space bar to return to the prompt.

## Set Local Terminal Parameters

The **terminal EXEC** 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 see a list of the commands for setting terminal parameters for the current session, use the following command in user EXEC mode:

Command	Purpose
terminal ?	List the commands for setting terminal parameters.

The following example shows the type of output **terminal ?** could generate:

```
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. The commands will be in two forms. The basic form will be in line configuration mode and can be saved permanently so that all terminal sessions are affected. The **terminal** form of the command is entered in EXEC mode and only affects the current session.

## Save Local Settings between Sessions

You can configure the Cisco IOS software to save local parameters set with **terminal EXEC** commands between sessions. Saving 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
<b>private</b>	Save local settings between sessions.

By default, user-set terminal parameters are cleared when the session ends with either the **exit EXEC** command, or when the interval set with the **exec-timeout** line configuration command has passed.

## End a Session

To end a session, use the following command:

Command	Purpose
<code>quit</code>	Use the quit EXEC command.

Refer to the “Managing Connections and System Banners” chapter for more information on exiting sessions and closing connections.

## Change 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 EXEC **terminal** 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 more common changes to the terminal and line settings:

- Define Escape Character and Other Key Sequences
- Specify Telnet Operation Characteristics
- Configure Data Transparency for File Transfers
- Specify an International Character Display

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

- Set Character Padding
- Specify the Terminal and Keyboard Type
- Change the Terminal Screen Length and Width
- Change Pending Output Notification
- Create Character and Packet Dispatch Sequences
- Display Debug Messages on the Console and Terminals
- Change Flow Control for the Current Session
- Set a Terminal-Locking Mechanism
- Configure Automatic Baud Rate Detection
- Set a Line as Insecure
- Configure Communication Parameters for Terminal Ports

## 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.

## Globally Define Escape Character and Other Key Sequences

To define or change the default key sequence, use one or more of the following commands in line configuration mode:

Command	Purpose
<b>escape-character</b> <i>ascii-number</i>	Change the system escape sequence. The escape sequence indicates that the codes that follow have special meaning. The default escape sequence is Ctrl-^. <sup>1</sup>
<b>activation-character</b> <i>ascii-number</i>	Define a session activation sequence or character. Entering this sequence at a vacant terminal begins a terminal session. The default activation sequence is the Return key.
<b>disconnect-character</b> <i>ascii-number</i>	Define the session disconnect sequence or character. Entering this sequence at a terminal ends the session with the router. There is no default disconnect sequence.
<b>hold-character</b> <i>ascii-number</i>	Define the hold sequence or character that causes output to the terminal screen to pause. To continue the output, enter any character after the hold character. To use the hold character in normal communications, precede it with the escape character. There is no default sequence.

1 Pressing **Ctrl** displays a caret (^) character. The escape sequence is **Ctrl-Shift-6**.

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.

## Define 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 one or more of the following commands in EXEC mode:

Command	Purpose
<b>terminal escape-character</b> <i>ascii-number</i>	Change the system escape sequence for the current session. The escape sequence indicates that the codes that follow have special meaning. The default sequence is Ctrl-^.
<b>terminal hold-character</b> <i>ascii-number</i>	Define the hold sequence or character that causes output to the terminal screen to pause for this session. There is no default sequence. To continue the output, type any character after the hold character. To use the hold character in normal communications, precede it with the escape character. You cannot suspend output on the console terminal.

The **terminal escape-character** command is useful, for example, if you have the default escape character defined for a different purpose in your keyboard file. Entering the escape character followed by the X key returns you to EXEC mode when you are connected to another computer.

## Specify Telnet Operation Characteristics

The following sections discuss telnet operation characteristics tasks:

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

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

To cause the router to generate a hardware Break signal on the EIA/TIA-232 line that is associated with a reverse Telnet connection for the current line and session, use the following command in EXEC mode:

---

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

---

The hardware Break signal occurs when a Telnet Interrupt-Process command is received on that connection. This command can be used to control the translation of Telnet IP commands into X.25 Break indications.

This command is also a useful workaround in the following situations:

- Several user Telnet programs send an Interrupt-Process command, but cannot send a Telnet Break signal.
- Some Telnet programs implement a Break signal that sends an Interrupt-Process command.

Some EIA/TIA-232 hardware devices use a hardware Break signal for various purposes. A hardware Break signal is generated when a Telnet Break command is received.

---

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

---

### Set the Line to Refuse Full-Duplex, Remote Echo Connections

You can set the line to allow the Cisco IOS software to refuse full-duplex, remote echo connection requests from the other end. This refusal suppresses negotiation of the Telnet Remote Echo and Suppress Go Ahead options. To set the current line to refuse to negotiate full-duplex for the current session, remote echo options on incoming connections, use the following command in EXEC mode:

---

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

---

---

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

---

## Allow 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
<b>terminal telnet speed</b> <i>default-speed</i> <i>maximum-speed</i>	Allow 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.

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

## Synchronize the Break Signal

You can set lines on the access server to cause a reverse Telnet line to send a Telnet Synchronize signal when it receives a Telnet Break signal. The TCP Synchronize signal clears the data path, but interprets incoming commands. To cause the Cisco IOS software to send a Telnet Synchronize signal when it receives a Telnet Break signal on the current line and session, use the following command in EXEC mode:

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

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

## Change the End-of-Line Character

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

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

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

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

## Configure 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, **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 successful 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, or CrossTalk can download a file across a terminal line. To temporarily configure a line to act as a pipe for file transfers, use the following command in EXEC mode:

Command	Purpose
<b>terminal download</b>	Set up 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**

## Specify 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**.

---

---

**Note** 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.

---

### Specify the International Character Display for All Lines

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

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

### Specify the International Character Display on a Hardware, Software, or Per-line Basis

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

### Specify an International 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 appropriate commands in EXEC mode:

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

### Set Character Padding

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

#### Globally Set Character Padding

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

Command	Purpose
<b>padding</b> <i>ascii-number count</i>	Set padding on a specific output character for the specified line.

#### Change 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
<b>terminal padding</b> <i>ascii-number count</i>	Set padding on a specific output character for the specified line for this session.

### Specify 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.

#### Globally Specify the Terminal Type

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

Command	Purpose
<b>terminal-type</b> { <i>terminal-name</i>   <i>terminal-type</i> }	Specify the terminal type.

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

#### Change 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
<b>terminal terminal-type</b> <i>terminal-type</i>	Specify the terminal type for this session.

Indicate the terminal type if it is different from the default of VT100. This default is used by TN3270 for display management and by Telnet and rlogin to inform the remote host of the terminal type.

To specify the current keyboard type for a session, use the following command in EXEC mode:

Command	Purpose
<b>terminal keymap-type</b> <i>keymap-name</i>	Specify the keyboard type for this 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 and give you their names.

## Change 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 a zero.

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

### Globally Change the Terminal Screen Length and Width

To set the terminal screen length and width, use the following commands in line configuration mode:

Step	Command	Purpose
1	<b>length</b> <i>screen-length</i>	Set the screen length.
2	<b>width</b> <i>characters</i>	Set the screen width.

### Change 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 one of the following commands in EXEC mode:

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

## Change 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 active one. For example, you might want to know when another connection receives mail or a message.

### Globally Set Pending Output Notification

To set pending output notification, use the following command in line configuration mode:

Command	Purpose
<b>notify</b>	Set up a line to notify a user of pending output.

### Set 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
terminal notify	Set up a line to notify a user of pending output for the current session.

### Create Character and Packet Dispatch Sequences

The Cisco IOS software supports dispatch sequences and TCP state machines that transmit data packets only when they receive a 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 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 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.

### Set Character and Packet Dispatch Sequences for a Line

Use the following commands in line configuration mode, as needed, for your particular system needs:

Step	Command	Purpose
1	<b>state-machine</b> <i>name state firstchar lastchar [nextstate   transmit]</i>	Specify the transition criteria for the states in a TCP state machine.
2	<b>dispatch-machine</b> <i>name</i>	Specify the state machine for TCP packet dispatch.
3	<b>dispatch-character</b> <i>ascii-number [ASCII-number2 . . . ascii-number]</i>	Define a character that triggers packet transmission.
4	<b>dispatch-timeout</b> <i>milliseconds</i>	Set the dispatch timer.

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

## Display 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
<b>terminal monitor</b>	Display 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 see the debugging messages.

## Change Flow Control for the Current Session

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

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

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

For more information about setting flow control or to set flow control on a line for more than the current session, refer to the “Configuring Modem Support and Asynchronous Devices” chapter in the *Dial Solutions Configuration Guide*.

## Set a Terminal-Locking Mechanism

You can enable a terminal-locking mechanism that allows a terminal to be temporarily locked by using the following command in line configuration mode:

Command	Purpose
<b>lockable</b>	Enable a temporary terminal locking mechanism.

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

## Configure Automatic Baud Rate Detection

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

Command	Purpose
<b>autobaud</b>	Set the terminal 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.

## Set a Line as Insecure

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

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

Command	Purpose
<b>insecure</b>	Set the line as a dial-up line.

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

## Configure Communication Parameters for Terminal Ports

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

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

## Record the 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
<b>location</b> <i>text</i>	Record the location of a serial device.

## Change 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
<b>terminal-queue entry-retry-interval</b> <i>interval</i>	Change the retry interval for a terminal port queue.

## LPD Protocol Support

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 successful 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
<b>printer</b> <i>printername</i> { <i>line number</i>   <i>rotary number</i> } [ <b>newline-convert</b> ]	Configure printer and specify 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 Saturn on the host Memphis:

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

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
show printer	List currently defined printers and their 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 */etc/printcap* on your UNIX system to include the definition of the remote printer in the Cisco IOS software.