



# Configuring Dial-In Terminal Services

---

This chapter describes how to configure support for asynchronous character stream calls running Telnet, rlogin, local-area transport (LAT), XRemote, or TN3270. It includes the following main sections:

- [Dial-In Terminal Service Overview](#)
- [Configuring Telnet and rlogin](#)
- [Telnet and rlogin Configuration Task List](#)
- [Using Cisco DialOut for Telnet Connections](#)
- [Connecting a VMS Host Using LAT](#)
- [LAT Configuration Task List](#)
- [Monitoring and Maintaining LAT Connections](#)
- [LAT Configuration and Connection Examples](#)
- [Configuring TN3270](#)
- [TN3270 Configuration Task List](#)
- [TN3270 Configuration and Connection Examples](#)
- [Configuring XRemote](#)
- [XRemote Configuration Task List](#)
- [XRemote Configuration and Connection Examples](#)

For a complete description of the dial-in terminal services commands in this chapter, refer to the [Cisco IOS Terminal Services Command Reference](#). To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

## Dial-In Terminal Service Overview

Inbound asynchronous character stream calls are routed to virtual terminal lines and virtual asynchronous interfaces, which are used to terminate incoming character streams that do not share a physical connection with the access server or router (such as a physical interface). A virtual asynchronous interface is the place where inbound Telnet, LAT, V.120, TN3270, and packet assembler/disassembler (PAD) calls or sessions terminate on the router. Virtual terminal lines are used for attaching to the router in a nonphysical way.



Configuring support for terminal service connections means enabling network devices running the same protocol to connect across a LAN or WAN through network and terminal-emulation software.

The following sections describe how to configure these supported dial-in terminal services:

- [Configuring Telnet and rlogin](#)—Of all protocol suites, TCP/IP is the most widely implemented on networks of all media types. TCP/IP is the current standard for internetworking and is supported by most computer vendors, including all UNIX-based workstation manufacturers. TCP/IP includes Telnet and rlogin.
- [Connecting a VMS Host Using LAT](#)—The proprietary LAT terminal connection protocol from Digital Equipment Corporation used with Digital minicomputers.
- [Configuring TN3270](#)—IBM 3278 terminal emulation provides TN3270-based connectivity to IBM hosts over serial lines.
- [Configuring XRemote](#)—The X Window Systems terminal protocol from Network Control Devices, Inc., provides network functionality to remote X terminals.

Each section provides examples of how to configure and connect to a terminal service.

## Configuring Telnet and rlogin

Telnet and rlogin are protocols that enable TCP/IP connections to a host. Telnet, a virtual terminal protocol that is part of the TCP/IP protocol suite, is the more widely used protocol. The rlogin protocol is a remote login service developed for the Berkeley Software Distribution (BSD) UNIX system. It provides better control and output suppression than Telnet, but can only be used when the host (typically, a UNIX system) supports rlogin. The Cisco IOS implementation of rlogin does not subscribe to the rlogin “trusted host” model. That is, a user cannot automatically log in to a UNIX system from the router, but must provide a user ID and a password for each connection.

Telnet allows a user at one site to establish a TCP connection to a login server at another site, then passes the keystrokes from one system to the other. Telnet can accept either an IP address or a domain name as the remote system address. In short, Telnet offers three main services:

- Network virtual terminal connection
- Option negotiation
- Symmetric connection

The Cisco implementation of Telnet supports the following Telnet options:

- Remote echo
- Binary transmission
- Suppress go ahead
- Timing mark
- Terminal type
- Send location
- Terminal speed
- Remote flow control
- X display location

# Telnet and rlogin Configuration Task List

To configure Telnet and rlogin, perform the tasks in the following sections:

- [Configuring Telnet and UNIX rlogin](#) (Required for Service)
- [Making Telnet and UNIX rlogin Connections](#) (Required for Making Connections)
- [Using UNIX Style Syntax for rlogin Connections](#) (Optional)

The section “[Monitoring TCP/IP Connections](#)” later in this chapter provides tasks for maintaining TCP/IP connections.

## Configuring Telnet and UNIX rlogin

To configure support for Telnet or rlogin calls, use the following commands beginning in line configuration mode.

Command	Purpose
Router(config-line)# <b>telnet speed</b> <i>default-speed maximum-speed</i>	Negotiates speeds on reverse Telnet lines.
Router(config-line)# <b>telnet refuse-negotiations</b>	Causes Telnet to refuse to negotiate full-duplex, remote echo requests on incoming connections.
Router(config-line)# <b>telnet transparent</b>	Sets line to send a RETURN (CR) as a CR followed by a NULL instead of a CR followed by a LINE FEED (LF).
Router(config-line)# <b>telnet sync-on-break</b>	Sets the line to send a Telnet SYNCHRONIZE signal when it receives a Telnet BREAK signal.
Router(config-line)# <b>telnet break-on-ip</b>	Sets the line to cause the system to generate a hardware BREAK signal on the EIA/TIA-232 line that is associated with a reverse Telnet connection when a Telnet Interrupt-Process command is received on that connection.
Router(config)# <b>ip tcp chunk-size</b> <i>number</i>	In global configuration mode, optimizes the line by setting the number of characters output before the interrupt executes.
Router(config-if)# <b>ip alias</b> <i>ip-address tcp-port</i>	In interface configuration mode, assigns an IP address to the service provided on a TCP port.
Router(config)# <b>busy-message</b> <i>hostname d message d</i>	In global configuration mode, defines a message that the router displays whenever a Telnet or rlogin connection to the specified host fails.
Router(config)# <b>login-string</b> <i>hostname d message [%secp] [%secw] [%b] d [%am] d</i>	In global configuration mode, defines a message that the router displays whenever a Telnet or rlogin connection to the specified host succeeds.
Router(config-line)# <b>notify</b>	Sets up a line to notify a user that has multiple, concurrent Telnet connections when output is pending on a connection other than the current one.
Router(config-line)# <b>refuse-message</b> <i>d message d</i>	Defines a “line-in-use” message to indicate that the line is currently busy.

The **telnet speed** command sets the line speed to match line speeds on remote systems in reverse Telnet, on host machines hooked up to an access server or router to access the network, or on a group of console lines hooked up to the access server or router 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.

The **telnet refuse-negotiations** command suppresses negotiation of the Telnet Remote Echo and Suppress Go Ahead options.

The **telnet transparent** command is useful for coping with different interpretations of end-of-line handling in the Telnet protocol specification.

The **telnet sync-on-break** command sets the line to cause a reverse Telnet line to send a Telnet SYNCHRONIZE signal when it receives a Telnet BREAK signal. The Telnet SYNCHRONIZE signal clears the data path, but the line still interprets incoming commands.

Enter the **telnet break-on-ip** command to control the translation of Telnet Interrupt-Process commands into X.25 BREAK indications, and to work around the following situations:

- Several user Telnet programs send a Telnet Interrupt-Process command, but cannot send a Telnet BREAK signal.
- Some Telnet programs implement a BREAK signal that sends a Telnet Interrupt-Process command.
- Some EIA/TIA-232 hardware devices use a hardware BREAK signal for various purposes.

When the **telnet break-on-ip** command is used with a correctly operating host, Cisco IOS software implements the Telnet SYNCHRONIZE and ABORT OUTPUT signals, which can stop output within one packet worth of data from the time the user types the interrupt character. Enter the **ip tcp chunk-size** command to configure a faster response to user interrupt characters. Changing the number of characters output, or chunk size, affects neither the size of the packet used nor the TCP window size, either of which would cause serious efficiency problems for the remote host and for the access server or router. Instead, the system software checks the Telnet status after the number of characters specified, causing only a relatively minor performance loss.

Use the **ip alias** command to configure connections to an IP address to act identically to connections made to the primary IP address of the server on the TCP port. A user trying to connect is connected to the first free line in a rotary group using the Telnet protocol.

With the **login-string** command options, you can set a pause, prevent a user from issuing commands during a pause, send a BREAK character, and use a percent sign (%) in the login string. The **busy-message** command and **login-string** command are only useful with two-step protocol translation sessions. For more information about protocol translation, see the chapter “Configuring Protocol Translation and Virtual Asynchronous Devices” in this publication.

For actual sample configurations on how to configure Telnet and rlogin, see the section “[Telnet and rlogin Examples](#)” later in this chapter.

## Making Telnet and UNIX rlogin Connections

To provide Telnet and rlogin connection capabilities, use the following commands in EXEC mode:

	Command	Purpose
Step 1	Router> <b>connect</b> <i>host</i> [ <i>port</i> ] [ <i>keyword</i> ] or Router> <b>telnet</b> <i>host</i> [ <i>port</i> ] [ <i>keyword</i> ]	Logs in to a host that supports Telnet. Refer to the descriptions for the <b>connect</b> and <b>telnet</b> commands in the <a href="#">Cisco IOS Terminal Services Command Reference</a> , for a list of supported keywords. <sup>1</sup>
Step 2	Router> <b>show hosts</b>	Displays a list of available hosts.
Step 3	Router> <b>show tcp</b>	Displays the status of all TCP connections.
Step 4	<b>ctrl^</b>	Logs out of the host by entering the default escape sequence. <sup>2</sup>
Step 5	Choose from the following list of escape sequences, according to your task:  Press <b>Ctrl^ b</b> if your task is to break. Press <b>Ctrl^ c</b> if your task is to interrupt a process (IP). Press <b>Ctrl^ h</b> if your task is to erase a character (EC). Press <b>Ctrl^ o</b> if your task is to abort an output display (AO). Press <b>Ctrl^ t</b> if your task is to confirm you are at the host. Press <b>Ctrl^ u</b> if your task is to erase a line (EL).	Logs out of the host by entering a special escape sequence. <sup>2</sup> These special Telnet sequences map generic terminal control functions to operating system-specific functions.
Step 6	<b>ctrl^ ?</b>	Lists the available Telnet commands at any time during the active Telnet session. <sup>2</sup>
Step 7	<b>exit</b> or <b>logout</b>	Exits a Telnet or rlogin session.

1. Cisco IOS software provides a robust collection of connection options. The options allow for enhanced sessions allowing, for example, encrypted sessions, Kerberos login, and File Transfer Protocol and World Wide Web connections. Additionally, it is possible to suppress system messages, including IP addresses and server names, displayed during session connection and disconnection. This function allows transparent TCP connections and can be useful when an asynchronous tunnel connection is being made.
2. Press and hold the **Ctrl** and **Shift** keys while pressing the **6** key. You can enter the command character as you hold down the **Ctrl** key or with **Ctrl** released; you can enter the command characters as either uppercase or lowercase letters.

With the Cisco IOS implementation of TCP/IP, you are not required to enter the **connect** or **telnet** commands to establish a Telnet connection. You can just enter the learned host name as long as the host name is different from a command word for the router. Telnet must be the default (you can make it the default with the **transport preferred** command). Use the **show hosts** EXEC command to display a list of the available hosts. Use the **show tcp** EXEC command to display the status of all TCP connections. The Cisco IOS software assigns a logical name to each connection, and several commands use these names to identify connections. The logical name is the same as the host name, unless that name is already in use or you change the connection name with the **name-connection** EXEC command. If the name is already in use, the Cisco IOS software assigns a null name to the connection. For an example of making a Telnet connection, see the section “[Telnet and rlogin Examples](#)” later in this chapter.

After you enter the **rlogin** command, you can have several concurrent rlogin connections open and switch between them. To open a new connection, exit the current connection by entering the escape sequence (**Ctrl-Shift-6** then **x** [**Ctrl^x**] by default) to return to the system command prompt, then open a new connection. For an example of making an rlogin connection or switching between connections, see the sections “[rlogin Connection Example](#)” or “[Switch Between Telnet and rlogin Sessions Example](#)” later in this chapter.

**Note**

We recommend that you use Encrypted Kerberized Telnet whenever you establish a Telnet session to a router or access server, which protects the integrity of the device. For information about Encrypted Kerberized Telnet, refer to *Cisco IOS Security Configuration Guide*.

## Using UNIX Style Syntax for rlogin Connections

The **rlogin** command supports the standard BSD UNIX **-l** option. Before this addition was introduced, the **rlogin** command allowed remote users to log in using the **/user username** option, which was not compatible with the standard UNIX **rlogin -l username** option.

This feature is supported on all of Cisco TCP/IP-enabled routers and access servers.

To set up this UNIX feature, use one of the following the following commands in EXEC mode:

Command	Purpose
Router# <b>rlogin</b> <i>hostname</i>	Enters the name of the host to which you are connecting.
Router# <b>rlogin</b> <i>hostname</i> [-l <i>hostname</i> ] [/user <i>hostname</i> ]	Enters the user name.
Router# <b>rlogin</b> <i>hostname</i> [-l <i>hostname</i> ] [/user <i>hostname</i> ] <b>debug</b>	(Optional) Enters the <b>debug</b> mode to troubleshoot the connection from the remote site to the host.
Router# <b>rlogin</b> <i>hostname</i> [-l <i>hostname</i> ] [/user <i>hostname</i> ] <b>/quiet</b>	(Optional) Enters the <b>/quiet</b> keyword to make a transparent connection from the remote site to the host.

When you are done with the UNIX session, use the **exit** command to end it.

## Monitoring TCP/IP Connections

To display the status of a TCP connection or view a summary of the TCP connection endpoints in the system, use the following commands in user EXEC mode:

Command	Purpose
Router> <b>show tcp</b> [ <i>line-number</i> ]	Displays the status of a TCP connection.
Router> <b>show tcp brief</b> [ <i>all</i> ]	Displays a summary of the TCP connection endpoints in the system.

## Telnet and rlogin Examples

This section provides the following examples:

- [Telnet Connection Example](#)
- [Telnet Connection Without and With Messages Suppressed Example](#)
- [rlogin Connection Example](#)
- [rlogin UNIX-Style Syntax Example](#)

- [Switch Between Telnet and rlogin Sessions Example](#)
- [List Supported Telnet Commands Example](#)

## Telnet Connection Example

The following example establishes a telnet connection to a host named server1 and specifies vt100 as the terminal type for the session:

```
Router> telnet server1 /terminal-type vt100
```

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

```
Router> host1
```

## Telnet Connection Without and With Messages Suppressed Example

The following examples show how to suppress the onscreen messages displayed during login and logout of a Telnet session.

The following example shows the messages displayed when a connection is made *without* using the optional **/quiet** keyword with the **telnet EXEC** command to suppress messages from the operating system:

```
Router# telnet Server3
```

```
Translating "Server3"...domain server (172.18.89.42) [OK]
Trying Server3--Server3.cisco.com (172.18.89.42)... Open
Kerberos:          No default realm defined for Kerberos!
```

```
login: User2
```

```
Password:
```

```
    Welcome to OpenVMS VAX version V6.1 on node CRAW
    Last interactive login on Tuesday, 15-DEC-1998 11:01
    Last non-interactive login on Sunday,  3-JAN-1999 22:32
```

```
Server3) logout
```

```
    User2          logged out at  16-FEB-2000 09:38:27.85
```

```
[Connection to Server3 closed by foreign host]
```

```
Router#
```

The following example shows the limited messages displayed when connection is made using the optional **/quiet** keyword:

```
Router# telnet Server3 /quiet
```

```
login: User2
```

```
Password:
```

```
    Welcome to OpenVMS VAX version V6.1 on node CRAW
    Last interactive login on Tuesday, 15-DEC-1998 11:01
    Last non-interactive login on Sunday,  3-JAN-1999 22:32
```

```
Server3) logout
```

```
    User2          logged out at  16-FEB-2000 09:38:27.85
```

```
Router#
```

The **/quiet** keyword is useful for making transparent connections during asynchronous tunnel connections. The keyword can be used with any of the EXEC connection commands—**connect**, **telnet**, and **rlogin**.

**Note**

The Cisco IOS software offers the **ip telnet quiet** global configuration command, which also suppresses onscreen messages during Telnet connections. The **ip telnet quiet** command is set globally, and is useful to Internet service providers that want to permanently suppress onscreen system connection messages that often include information such as server names and IP addresses. Refer to the [Cisco IOS Dial Technologies Command Reference](#), for more information about the **ip telnet quiet** command.

## rlogin Connection Example

The following example makes an rlogin connection to a host at address 172.31.21.2 and enables the message mode for debugging:

```
Router> rlogin 172.31.21.2 debug
```

## rlogin UNIX-Style Syntax Example

The following example illustrates how a user named jsmith can use the **rlogin ?** help command and the debug mode to establish and troubleshoot a remote connection to the host named Alviso:

```
Router> rlogin ?
WORD IP address or hostname of a remote system
Router> rlogin Alviso ?
-1 Specify remote username
/user Specify remote username
debug Enable rlogin debugging output
<cr>
Router> rlogin Alviso -1 ?
WORD Remote user name
Router> rlogin Alviso -1 jsmith ?
debug Enable rlogin debugging output
<cr>
Router> rlogin Alviso -1 jsmith debug
```

## Switch Between Telnet and rlogin Sessions Example

You can switch between sessions by escaping one session and resuming a previously opened session. The following example shows how to escape out of a connection to the host named host1 and to resume connection 2. You escape out of the current session and return to the EXEC prompt by entering the command sequence **Ctrl-Shift-6** then **x**. Resume the connection with the **resume** command.

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

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

```
Router> 3
```

To list all the open sessions associated with the current terminal line, use the **where** command.

## List Supported Telnet Commands Example

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

```
Ctrl-^ ?
```



A sample of this list follows:

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

**Note**

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

## Using Cisco DialOut for Telnet Connections

The Cisco DialOut feature enables users on a workstation operating Windows to send faxes or connect to service provider services outside the LAN by using modems attached or internal to a network access server. The Cisco DialOut feature extends the functionality of Telnet by enabling users to control the activity of these modems from their desktop computers using standard communications software.

The Cisco DialOut feature has two components:

- Telnet Extensions for Dialout—Network access server component
- The DialOut Utility—Client/desktop component

Both components are required and neither can function as a stand-alone feature.

The Telnet Extensions for Dialout component uses reverse Telnet to access modems attached to the network access server. This component enables the network access server to interface with the client/desktop component of the Cisco DialOut feature and to return CARRIER DETECT signals to the communications software so that the software can determine when to start dialing a particular number.

Telnet extensions allow the communications software running on the desktop computer of the client to control modem settings such as baud rate, parity, bit size, and stop bits.

To enable this feature, you only need to configure the access server or router for reverse Telnet and configure the appropriate lines to send and receive calls.

The client/desktop component of Cisco DialOut feature must be installed on the client workstation before this feature can be used. For information about installing and using the client/desktop component of the Cisco Dial-Out feature, and configuring the access server, see the *DialOut Utility User Guide* Cisco publication at Cisco.com.

## Configuring Stream TCP

Stream TCP connections, or raw TCP or TCP-Clear connections as they are sometimes called, are used to transport a stream of 8-bit characters as-is over an IP network, between a TCP client and TCP server system. This method is used to transport legacy asynchronous application data through an IP network, for example, with a Point-of-Sale (PoS) terminal connecting to an application server.

To establish a Stream TCP connection from an EXEC session, use the **/stream** keyword with the **telnet** command. You will also generally want to configure the line to provide for data transparency. See the following procedure for the steps to do this.

## Stream TCP Autocommand Procedure

In the following procedure, a line is configured so that any connection into it is automatically connected using Stream TCP to the application server at the specified IP address and TCP port (IP address 10.1.2.3 and TCP port 4321 in the examples).

- Step 1** Configure the line for data transparency using the following configuration as an example:

```
Router# configure terminal

Router(config)# line 33
Router(config-line)# no motd-banner
Router(config-line)# no exec-banner
Router(config-line)# no vacant-message
Router(config-line)# escape-character NONE
Router(config-line)# no hold-character
```

- Step 2** Configure the autocommand:

```
Router(config-line)# autocommand telnet 10.1.2.3 4321 /quiet /stream
```

- Step 3** Configure the **telnet-faststream** option (this is an optional step). On platforms that support this feature such as the Cisco AS5800 access servers, you may want to configure the **telnet-faststream autocommand** option to provide for Stream TCP performance enhancements. An example of how this option can be entered follows:

```
Router(config-line)# autocommand-options telnet-faststream
```

## Connecting a VMS Host Using LAT

Connection to a VMS host is slightly different if you are connecting to a VMS host running VMS Version 5.4 or earlier than when connecting to a VMS host running VMS Version 5.5 or later software.

### VMS Version 5.4 or Earlier System

If a host-initiated connection is received that specifies a destination port number that corresponds to a virtual port on the router, a virtual EXEC process will be created to allow the user to log in. This process can be used, in conjunction with the Digital **set host/dte** command on VMS, to connect to a router named router1 from a VMS host node, as shown in the following example:

```
$lcp ==$latcp
$lcp create port lta300:
$lcp set port lta300:/service=able /node=router1
$set host/dte lta300:
```

## VMS Version 5.5 or Later System

To connect to a VMS host running VMS Version 5.5 or later software, you must turn on the outgoing connections of the VMS LAT hosts and use the Digital **set host/lat** command, as shown in the following example:

```
$lcp ::= $latcp
$lcp set node/connection =outgoing
$set host/lat able
```

## Port Names When Configuring a LAT Printer

When you configure a LAT printer, the LAT port name is the line number without a “TTY” designation on the **show lines** command output. For example, if you configure terminal line 10 (named ABLE) to be a LAT printer port, you must use the OpenVMS command to associate an arbitrary LAT device to the LAT port name, as follows:

```
$lcp ::= $lcp
$lcp create port lta300:
$lcp set port/node=ABLE/port=10 lta300:
```

The LAT port name is the line number without the “TTY,” regardless of whether the format of the TTY line number is decimal or octal.

## Additional LAT Capability

The Cisco IOS software fully supports the LAT protocol suite, and provides the following features:

- High-speed buffering—Handles a full screen of data (2000 characters) at full speed without requiring additional flow control.
- Protocol transparency—Handles connections transparently. The user needs no protocol information to establish a connection.
- Simplified configuration management—Uses logical names for LAT group codes to simplify the network structure.
- Maintenance Operation Protocol (MOP)—Supports the Digital protocol to support the request ID message, periodic system ID messages, and the remote console carrier functions for Ethernet interfaces.

## LAT Configuration Task List

The Cisco IOS software LAT protocol is supplied with a default configuration and does not require additional configuration for you to use it.

To enable LAT and customize LAT for your particular network environment, perform the tasks described in the following sections:

- [Configuring Basic LAT Services](#) (Required for Service)
- [Enabling Inbound Services](#) (As Required)
- [Controlling Service Announcements and Service Solicitation](#) (As Required)
- [Configuring Traffic Timers](#) (As Required)

- [Optimizing Performance](#) (As Required)
- [Defining LAT Access Lists](#) (As Required)
- [Enabling Remote LAT Modification](#) (As Required)
- [Making LAT Connections](#) (Required for Making Connections)

The section “[Monitoring and Maintaining LAT Connections](#)” later in this chapter provides tips for maintaining LAT connections. The section “[LAT Configuration and Connection Examples](#)” later in this chapter provides LAT configuration examples.

## Configuring Basic LAT Services

To enable basic LAT services, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	Router(config-if)# <b>lat enabled</b>	Enables the LAT protocol. LAT is disabled by default.
Step 2	Router(config-if)# <b>lat node</b> <i>node-name</i>	Gives the router a LAT node name that is different than the host name.
Step 3	Router(config-line)# <b>lat out-group</b> { <i>groupname</i>   <i>number</i>   <i>range</i>   <b>all</b> }	(Optional) Defines the group list for an outgoing connection on a specified line.
Step 4	Router(config)# <b>lat group-list</b> <i>groupname</i> { <i>number</i>   <i>range</i>   <b>all</b> } [ <b>enabled</b>   <b>disabled</b> ]	(Optional) Specifies logical names for group lists.
Step 5	Router(config)# <b>lat service-group</b> { <i>groupname</i>   <i>number</i>   <i>range</i>   <b>all</b> } [ <b>enabled</b>   <b>disabled</b> ]	(Optional) Specifies groups to be advertised.
Step 6	Router(config-line)# <b>lat remote-modification</b>	(Optional) Enables remote LAT modification of line characteristics.

Use the **lat out-group** command to define the list of services to which a user can connect. You create this list by defining the group code lists used for connections from specific lines. You can limit the connection choices for an individual line by defining the group code lists for an outgoing connection. When a user initiates a connection with a LAT host, the line of the user must share a common group number with the remote LAT host before a connection can be made.

Use the **lat group-list** command to specify a name for group lists to simplify the task of entering individual group codes. A name makes it easier to refer to a long list of group code numbers. To display the defined groups, use the **show lat groups** command.

Use the **lat service-group** command to specify a group code mask to use when advertising all services for a node. You can enter more than one group code by listing the numbers. You can also enter both a group code name and group codes.

Use the **lat remote-modification** line configuration command to configure a LAT line so that a remote LAT node can change the operating characteristics of the line.

## Enabling Inbound Services

Just as LAT services are offered by host computers, they also can be offered by access servers and routers, because they implement both the host and server portions of the LAT protocol. This capability allows connections from either hosts or local access servers or routers. A host connected to a local device is called a *host-initiated connection*.

The tasks described in this section define support for host-initiated connections. This support includes refining the list of services that the router will support. An incoming session can be to either a port or a service. The port name is the terminal line number, as reported by the **show users all EXEC** command.

To enable inbound services, use the following commands in global configuration mode as needed:

Command	Purpose
Router(config)# <b>lat service</b> <i>service-name</i> <b>password</b> <i>password</i>	Sets the LAT password for a service.
Router(config)# <b>lat service</b> <i>service-name</i> <b>ident</b> <i>identification</i>	Sets the LAT service ID for a specific service.
Router(config)# <b>lat service</b> <i>service-name</i> <b>rating</b> <i>static-rating</i>	Specifies a static service rating for a specific service.
Router(config)# <b>lat service</b> <i>service-name</i> <b>rotary</b> <i>group</i>	Configures a LAT rotary group.
Router(config)# <b>lat service</b> <i>service-name</i> <b>autocommand</b> <i>command</i>	Associates a command with a specific service for auto-execution.
Router(config)# <b>lat service</b> <i>service-name</i> <b>enabled</b>	Enables inbound connections to a specific service.

Use the **show lat advertised EXEC** command to display LAT services offered to other systems on the network.

A service must be specifically enabled, but not all of the attributes in the previous task table are necessary in a particular environment.

## Controlling Service Announcements and Service Solicitation

You can configure the Cisco IOS software to support the service responder feature that is part of the LAT Version 5.2 specification.

Specifically, the DECserver90L+, which has less memory than other Digital servers, does not maintain a cache of learned services. Instead, the DECserver90L+ solicits information about services as they are needed.

LAT Version 5.2 nodes can respond for themselves, but LAT Version 5.1 nodes, for example, VMS Version 5.4 or earlier nodes, cannot. Instead, a LAT Version 5.2 node configured as a service responder can respond in proxy for those LAT Version 5.1 nodes.

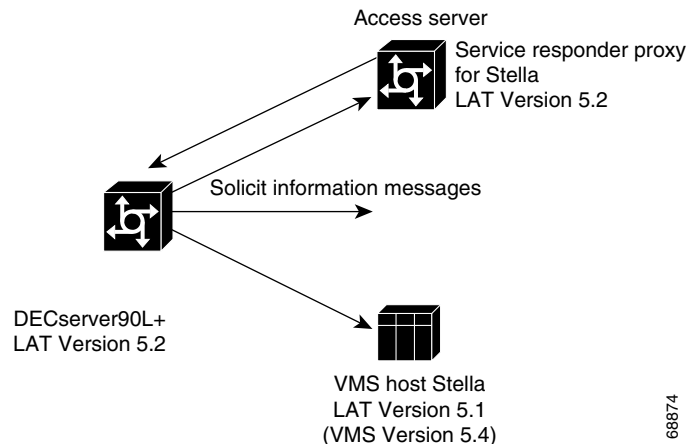
The Cisco IOS software can be configured as a LAT service responder. Of course, if all your nodes are LAT Version 5.2 nodes, you need not enable the service responder features.

To control service announcements and service solicitations, use the following commands in global configuration mode:

	Command	Purpose
<b>Step 1</b>	Router(config)# <b>lat service-responder</b>	Enables a proxy node to respond to solicit-information multicast messages.
<b>Step 2</b>	Router(config)# <b>no lat service-announcements</b>	Disables periodic broadcasts of service advertisements.
<b>Step 3</b>	Router(config)# <b>lat service-timer</b> <i>interval</i>	Adjusts the time between service announcements.

Use the **lat service-responder** command to configure the Cisco IOS software to respond to solicit information requests addressed to LAT Version 5.1 nodes. This function allows nodes that do not cache service advertisements to interoperate with nodes that do not respond to solicit requests. Figure 1 shows how a router can act as a proxy for LAT servers.

**Figure 1 Router as Proxy for LAT Server**



The DECserver90L+ broadcasts a solicit information request in search of service for address Stella. The VMS host, Stella, is unable to respond to the request because it is running LAT Version 5.1. The access server is running LAT Version 5.2 with service responder enabled and informs the DECserver90L+ of the address for Stella.

Use the **no lat service-announcements** command to disable periodic broadcasts of service announcements. If service announcements are enabled, the LAT node will periodically broadcast service advertisements. If service announcements are disabled, the LAT node will not send service announcements, so a remote node requiring connection to the local node must use solicit-information messages to look up node information. Disable service announcements only if all of the nodes on the LAN support the service responder feature.

Use the **lat service-timer** command to adjust the time between LAT service advertisements for services offered. This command is useful in large networks with many LAT services and limited bandwidth.

## Configuring Traffic Timers

You can customize the environment for sending LAT messages. The Cisco IOS implementation of LAT allows you to set the following features:

- The number of retransmissions before declaring a system unreachable
- The interval of time LAT waits before sending a keepalive message on an idle connection
- The interval of time LAT waits between transmission of messages

These features affect all LAT connection types.

To enable these features, use the following commands in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>lat retransmit-limit</b> <i>number</i>	Sets the message retransmit limit.
Step 2	Router(config)# <b>lat ka-timer</b> <i>seconds</i>	Sets the keepalive timer.
Step 3	Router(config)# <b>lat vc-timer</b> <i>milliseconds</i>	Sets the virtual circuit timer.

## Optimizing Performance

To optimize performance for your LAT environment, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>lat vc-sessions</b> <i>number</i>	Sets the maximum number of sessions on a LAT virtual circuit. The maximum (and default) number of sessions is 255.
Step 2	Router(config)# <b>lat host-buffers</b> <i>receive-buffers</i>	Allows a LAT host node to receive more than one message at a time.
Step 3	Router(config)# <b>lat server-buffers</b> <i>receive-buffers</i>	Allows a LAT server node to receive more than one message at a time.
Step 4	Router(config)# <b>lat host-delay</b> <i>number</i>	Specifies the delay acknowledgment for incoming LAT slave connections, where <i>number</i> is milliseconds.

Use the **lat host-buffers** command to set the number of messages received by a host at one time. Increasing this number can enhance performance. Before LAT Version 5.2, LAT allowed only one outstanding message at one time on a virtual circuit. This restriction could limit the performance of the Cisco IOS software when it processed a large number of messages because only one Ethernet packet of data could be in transit at a time. During virtual circuit startup, each side communicates to the other how many outstanding messages it is willing to accept.

Use the **lat server-buffers** command to set the number of messages received by a server at one time. Increasing this number can enhance performance. Before LAT Version 5.2, LAT allowed only one outstanding message at one time on a virtual circuit. This restriction could limit the performance of Cisco IOS software when it processed a large number of messages because only one Ethernet packet of data could be in transit at a time. With LAT Version 5.2, nodes can indicate that they are willing to receive more than one message at a time. During virtual circuit startup, each side communicates to the other how many outstanding messages it is willing to accept.

Use the **lat host-delay** command to set a user-defined delay for the acknowledgment for incoming LAT slave connections. This command is useful in situations where you need to control the delay. For example, if data is being transferred between a Digital server (using LAT) and a UNIX host (using Telnet) via a protocol translator, the protocol translator imposes the LAT delay on the Telnet and the LAT service, where Telnet may time out due to the LAT restriction.

## Defining LAT Access Lists

Because LAT groups were not intended to implement security or access control, the Cisco IOS software supports *access lists* to provide these functions. An access list is a sequential collection of permit and deny conditions that serve to restrict access to or from LAT nodes on a specific terminal line. Each access list statement defines a permit or deny condition and a matching criterion for the node name.

When a LAT connection is attempted (either incoming or outgoing), the node name of the destination service (*not* the service name) is compared against the regular expression. If they match, the connection is permitted or denied as specified.

To define access lists and conditions, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router# <b>configure terminal</b>	Enters global configuration mode.
Step 2	Router(config)# <b>lat access-list</b> <i>number</i> { <b>permit</b>   <b>deny</b> } <i>node-name</i>	Specifies an access condition.
Step 3	Router(config)# <b>line</b> <i>line-number</i>	Enters line configuration mode.
Step 4	Router(config-line)# <b>access-class</b> <i>access-list-number</i> { <b>in</b>   <b>out</b> }	Restricts incoming and outgoing connections between a particular terminal line or group of lines and the node names in an access list.

## Enabling Remote LAT Modification

You can configure a LAT line so that a remote LAT node can change the operating characteristics of the line. To enable remote LAT modification, use the following command in line configuration mode:

Command	Purpose
Router(config-line)# <b>lat remote-modification</b>	Enables remote LAT modification of line characteristics.

## Making LAT Connections

The LAT protocol is most often used to connect routers to Digital hosts. LAT is a Digital-proprietary protocol, and the Cisco IOS software uses LAT technology licensed from Digital to allow the following LAT services:

- Make a LAT connection
- Define a group code list for outgoing LAT connections
- Switch between LAT sessions
- Use Digital commands on the server
- Exit a LAT session

For actual LAT connection examples, see the section [“LAT Configuration and Connection Examples”](#) later in this chapter.

To enable specific LAT connections or services, use the following commands in EXEC mode:



	Command	Purpose
<b>Step 1</b>	Router> <b>lat</b> <i>name</i> [ <b>node</b> <i>node-name</i>   <b>port</b> <i>portname</i>   / <b>debug</b> ]	Connects to a LAT host. <sup>1</sup>
<b>Step 2</b>	Router> <b>terminal lat out-group</b> { <i>groupname</i>   <i>number</i>   <i>range</i> }	(Optional) Defines a temporary list of services to which you or another user can connect by defining the group code lists used for connections from specific lines.
<b>Step 3</b>	Router> <b>show lat services</b> [ <i>service-name</i> ]	(Optional) Lists available LAT services.
<b>Step 4</b>	Router> <b>help</b>	(Optional) Lists the subset of Digital commands that the Cisco IOS software supports.

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

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

To specify a temporary list of services to which you or another user can connect, you must define the group code lists used for connections from specific lines. You limit the connection choices for an individual line by defining the group code lists for an outgoing connection. To define a group code list, use the **terminal lat out-group** command. When a user initiates a connection with a LAT host, the line of the user must share a common group number with the remote LAT host before a connection can be made. The group code range *must be* a subset of the configured group code range of the line.

You can have several concurrent LAT sessions open and switch between them. To open a subsequent session, first enter the escape sequence (Ctrl-Shift-6 then x [Ctrl^x] by default) to suspend the current session. Then open a new session. To list the available LAT services, enter the **show lat services EXEC** command.

When you are done with the LAT session, use the **exit** command to end it, then terminate the active LAT session by entering the Ctrl-C key sequence.

## Monitoring and Maintaining LAT Connections

To monitor and maintain LAT connections, use the following commands in EXEC mode as needed:

Command	Purpose
Router> <b>clear entry</b> <i>number</i>	Deletes an entry from the queue.
Router> <b>show entry</b>	Displays queued host-initiated connections.
Router> <b>show lat advertised</b>	Displays LAT services offered to other LAT systems.
Router> <b>show lat groups</b>	Displays defined LAT groups.
Router> <b>show lat nodes</b>	Displays information about LAT nodes.

Command	Purpose
Router> <b>show lat services</b> [ <i>service-name</i> ]	Displays information about LAT learned services.
Router> <b>show lat sessions</b> [ <i>line-number</i> ]	Displays active LAT sessions.
Router> <b>show lat traffic</b>	Displays traffic and resource utilization statistics.
Router> <b>show node</b> [ <i>all</i>   <i>node-name</i> ] [ <i>counters</i>   <i>status</i>   <i>summary</i> ]	Displays information about LAT nodes. Information is displayed in the same way as in the Digital interface.
Router> <b>show service</b> [ <i>service-name</i> ]	Displays LAT learned services.

## LAT Configuration and Connection Examples

This section provides the following LAT examples:

- [Basic LAT Service Example](#)
- [LAT Service with Selected Group Codes Example](#)
- [Displaying LAT Services on the Same LAN Example](#)
- [Establishing an Outbound LAT Session Example](#)
- [Logically Partitioning LAT Services by Terminal Line Example](#)
- [LAT Rotary Groups Example](#)
- [Associating a Rotary Group with a Service Example](#)
- [LAT Access List Example](#)
- [LAT Connection Examples](#)

### Basic LAT Service Example

The following example establishes the LAT service named ABLE for your router. Subsequently, your router advertises ABLE (with default group code 0) on the LAN. Other LAT nodes can connect to you using LAT service ABLE, provided the group codes on the LAT nodes and the group codes for ABLE intersect. By default, most LAT nodes, such as OpenVMS Version 5.5 hosts, have user group code set to 0, so you have default access to ABLE.

```
! Create LAT service with password protection and
! identification string using the following global configuration commands.
lat service ABLE password secret
lat service ABLE ident Welcome to my machine
```

### LAT Service with Selected Group Codes Example

The following example establishes the LAT service named ABLE from your router with selected group codes 1, 4 through 7, and 167. This configuration limits inbound access to those LAT nodes that have group codes that intersect with those for LAT service ABLE.

```
! Establish a LAT group list.
lat group-list HUBS 1 4-7 167
```

```

!
! Enable LAT group list for the service-group.
lat service-group HUBS enabled
!
! Create LAT service with password protection and
! identification string.
lat service ABLE password secret
lat service ABLE ident Welcome to my machine

```

## Displaying LAT Services on the Same LAN Example

The following example demonstrates how you can check which LAT services are on the same LAN as your router. Note that the LAT service named ABLE is also listed, with the “Interface” column listing the interface as “Local.”

```
Router> show lat services
```

Service Name	Rating	Interface	Node (Address)
CAD	16	Ethernet0	WANDER
ABLE	16	Local	
CERTIFY	33	Ethernet0	STELLA

## Establishing an Outbound LAT Session Example

The following example establishes a LAT session to remote LAT service HELLO using an interactive session:

```
Router> lat HELLO
```

## Logically Partitioning LAT Services by Terminal Line Example

The following example illustrates how LAT services are logically partitioned by terminal line. At the example site, lines 1 through 7 go to the shop floor, lines 8 through 11 go to the Quality Assurance department, and lines 12 through 16 go to a common area.

```

! Define LAT groupnames.
lat group-list DEFAULT 0
lat group-list FLOOR 3
lat group-list QA 4

line 1 7
lat out-group FLOOR enabled
lat out-group DEFAULT disabled
line 8 11
lat out-group QA enabled
lat out-group DEFAULT disabled
line 12 16
lat out-group DEFAULT QA FLOOR enabled

```

## LAT Rotary Groups Example

The following example illustrates how to configure a range of lines for rotary connections and then establishes the LAT service named Modems for rotary connection:

```
! Establish rotary groups.
```

```

line 3 7
  rotary 1
!
! Establish modem rotary service.
!
  lat service Modems rotary 1
  lat service Modems enabled

```

## Associating a Rotary Group with a Service Example

The following example defines a service that communicates with a specific line and defines a rotary with only that line specified. You can establish rotary groups using line configuration commands and the **rotary** line configuration command.

```

hostname ciscots
! Service name for the access server as a whole.
lat service ciscopt enable
! Set up some lines with unique service names.
line 1
  rotary 1
  lat service ciscopt1 rotary 1
  lat service ciscopt1 enable
!
line 2
  rotary 2
  lat service ciscopt2 rotary 2
  lat service ciscopt2 enable

```

## LAT Access List Example

The following example illustrates incoming permit conditions for all IP hosts and LAT nodes with specific characters in their names and a deny condition for X.25 connections to a printer. Outgoing connections, however, are less restricted.

```

! Permit all IP hosts, LAT nodes beginning with "VMS" and no X.25
! connections to the printer on line 5.
!
access-list 1 permit 0.0.0.0 255.255.255.255
lat access-list 1 permit ^VMS.*
x29 access-list 1 deny .*
!
line 5
  access-class 1 in
!
! Meanwhile, permit outgoing connections to various places on all the
! other lines.
!
! Permit IP access within cisco.
access-list 2 permit 172.30.0.0 0.0.255.255
!
! Permit LAT access to the Stella/blue complexes.
lat access-list 2 permit ^STELLA$
lat access-list 2 permit ^BLUE$
!
! Permit X25 connections to infonet hosts only.
x29 access-list 2 permit ^31370
!
line 0 99
  access-class 2 out

```

The following example illustrates how to define access lists that permit all connections, thereby conforming to software behavior prior to Cisco IOS Release 9.0. Remember that the value supplied for the *list* argument in both variations of the **access-class** commands is used for *all* protocols supported by the Cisco IOS software. If you are already using an IP access list, it will be necessary to define LAT (and possibly X.25) access lists permitting connections to all devices, to emulate the behavior of earlier software versions.

```
access-list 1 permit 172.30.0.0 0.0.255.255
access-list 1 permit 172.30.0.0 0.0.255.255
!
line 1 40
  access-class 1 out
! Define LAT access list that permits all connections.
  lat access-list 1 permit .*
```

## LAT Connection Examples

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

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

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

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

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

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

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

The following example uses the LAT session debugging capability:

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

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

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

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

# Configuring TN3270

IBM 3270 display terminals are among the most widely implemented and emulated terminals for host-based computing in the computing community. Information in this section describes the TN3270 terminal emulation environment and how to use and create files that allow terminals connected to the access server or router to be used for TN3270 operation.

This section does not describe how to configure a TN3270 server. For information about configuring TN3270 server support in the Cisco IOS software, see the *Cisco IOS Bridging and IBM Networking Configuration Guide*.

The following sections are included:

- [TN3270 Overview](#)
- [TN3270 Configuration Task List](#)
- [TN3270 Configuration and Connection Examples](#)

## TN3270 Overview

TN3270 terminal emulation software allows any terminal to be used as an IBM 3270-type terminal. Users with non-3270 terminals can take advantage of the emulation capabilities to perform the functions of an IBM 3270-type terminal. The Cisco IOS software supports emulation of the following terminal types:

- IBM 3278-2 terminal with an 80-by-24 display
- IBM 3278-2 terminal with a 24-by-80 display
- IBM 3278-3 terminal with a 32-by-80 display
- IBM 3278-4 terminal with a 48-by-80 display
- IBM 3278-5 terminal with a 27-by-132 display

True IBM 3270-type terminals use a character format referred to as Extended Binary Coded Decimal Interchange Code (EBCDIC). EBCDIC consists of 8-bit coded characters and was originally developed by IBM. Emulation is made possible by the termcap protocol. Termcap functions translate the keyboard and terminal characteristics for ASCII-type terminals into those required for an IBM host.

Formally, a termcap is a two-part terminal-handling mechanism. It consists of a database and a subroutine library. The database describes the capabilities of each supported terminal, and the subroutine library allows programs to query the database and to make use of the values it contains. For more information about defining termcaps, refer to the commercially available book *termcap & terminfo*, by Jim Strang, Tim O'Reilly, and Linda Mui.

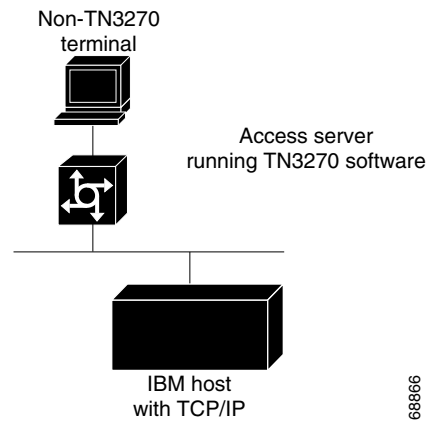
The Cisco IOS software includes a default termcap entry for Digital VT100 terminal emulation. More samples are available directly from Cisco at <http://www.cisco.com/warp/public/494/1.html>. This URL is subject to change without notice.

TN3270 emulation capability allows users to access an IBM host without using a special IBM server or a UNIX host acting as a server. (See [Figure 2](#).) The IBM host must directly support TCP/IP or have a front-end processor that supports TCP/IP.

A two-step translation method connects IBM hosts from LAT, TCP, and X.25/PAD environments. (See the chapter "Configuring Protocol Translation and Virtual Asynchronous Devices" later in this publication for more information about two-step translations.) In general, TN3270 support allows

outgoing TN3270 connections only. In other words, LAT, TCP, and X.25/PAD users must first establish a connection with the access server or router, then use the TN3270 facility from the Cisco IOS software to make a connection to the IBM host.

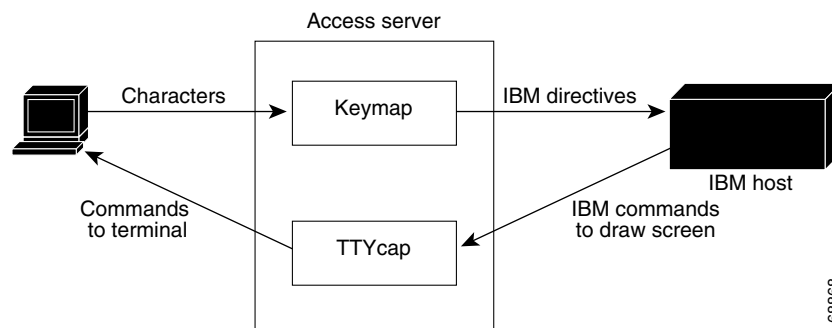
**Figure 2** Typical TN3270 Connection Environment



## Keymaps and ttycaps

Figure 3 shows how the keymapping and TTYcap functionality in the Cisco IOS software allows IBM hosts and non-IBM terminals to communicate.

**Figure 3** Keymaps and TTYcaps



Keymaps and TTYcaps have the following functionality:

- **Keymap**—Keyboard map file. Terminals send a key sequence for every key used to send packets to an IBM host. The keymapping function in the Cisco IOS software identifies special sequences and converts them to directives to the IBM host. A minimal level of keymapping is supported by default. Several keys can convert to the same IBM directives.
- **TTYcap**—Terminal emulation file. IBM devices and software send commands to the terminal, including cursor position, clear screen, and so on. The TTYcap functionality in the Cisco IOS software changes IBM directives into the terminal language. By default, protocol translation on access servers and routers conforms to the American National Standards Institute (ANSI) terminal standard, which is VTxxx terminal compatible.

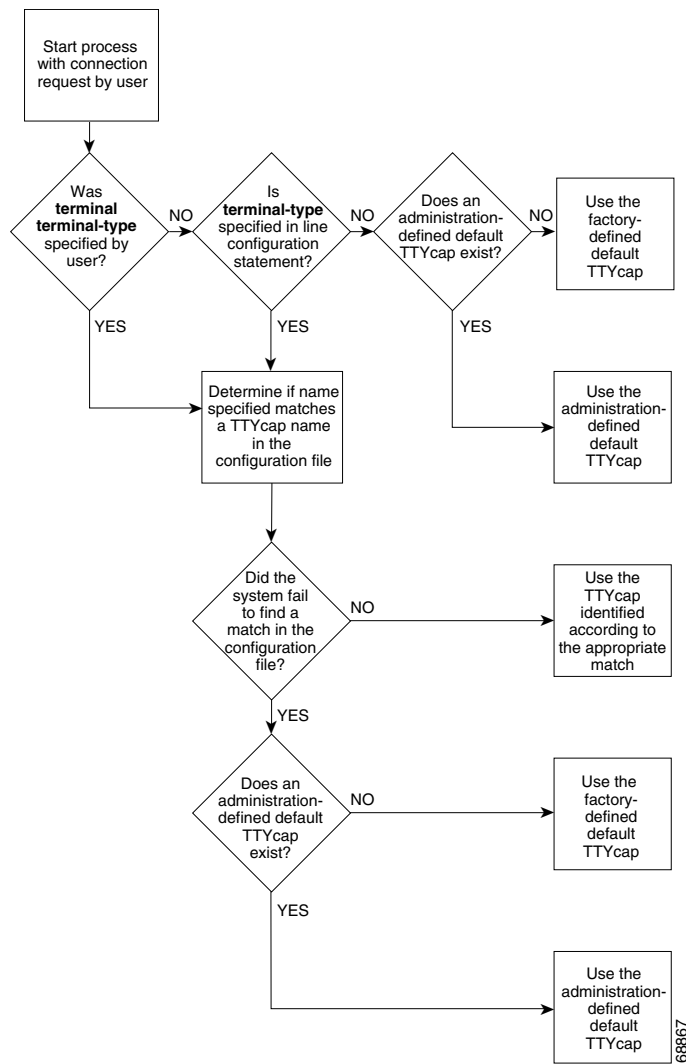
# Startup Sequence Priorities

At system startup, the Cisco IOS software uses the following decision sequence when selecting a TTYcap:

1. Use a user-supplied terminal emulation filename.
2. Use a terminal emulation filename specified using line configuration commands.
3. Use a default terminal emulation filename supplied by the administrator.
4. Use the default VT100 emulation.

Figure 4 illustrates the decision process used by the Cisco IOS software to choose a TTYcap for a specific TN3270 session.

**Figure 4** Decision Diagram for Cisco IOS Software TTYcap Selection Process





At system startup, the Cisco IOS software uses the following decision sequence when selecting a keymap:

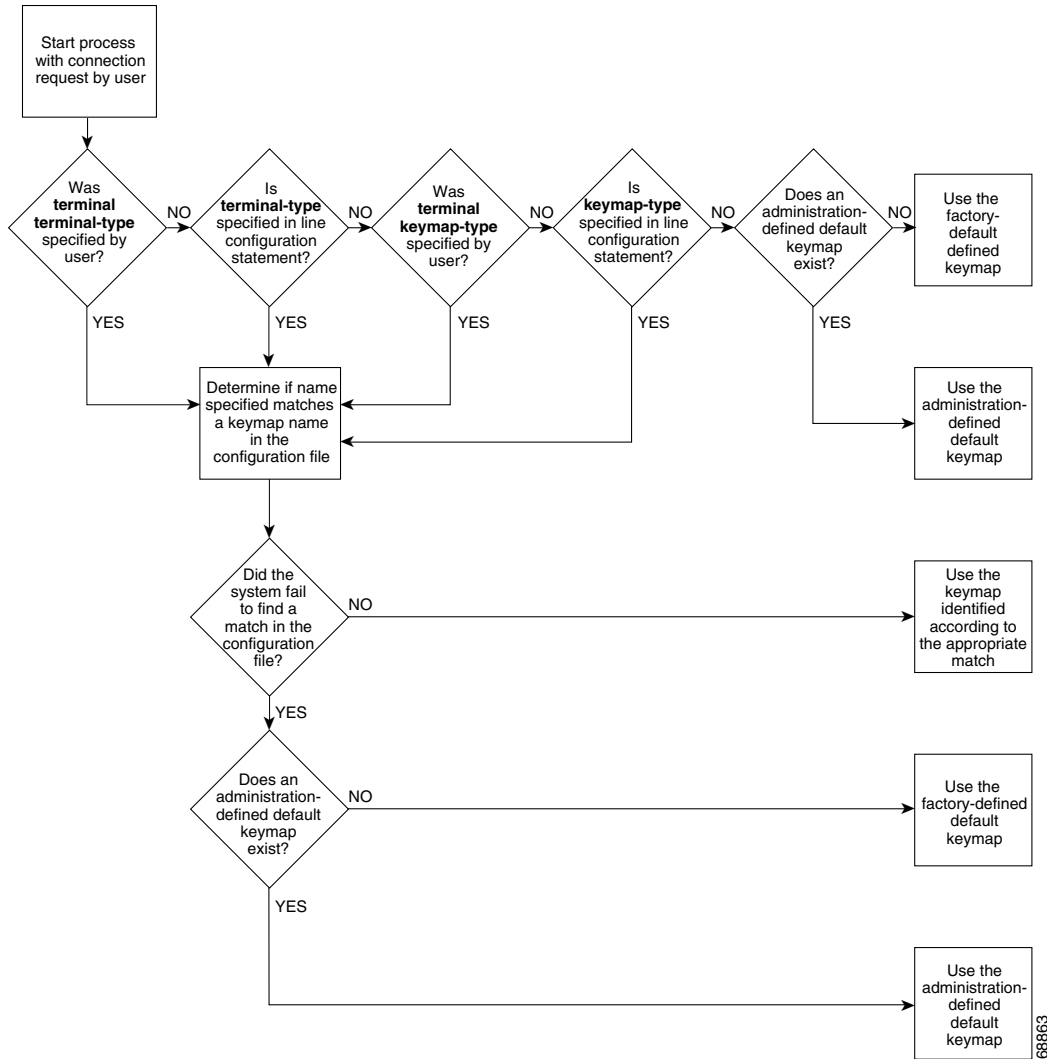
1. Use a user-supplied keyboard map filename.
2. Use a keyboard map filename specified using line configuration commands.
3. Use a user-supplied terminal emulation filename.
4. Use a terminal emulation filename specified using line configuration commands.
5. Use the default keyboard map filename supplied by the administrator.
6. Use the default VT100 emulation.

The software uses the following criteria to determine the file to use:

- If a filename is specified by the user but fails to match any name in the configuration file, the access server or router adopts the default specified by the administrator. If one has not been specifically defined, the factory-default emulation file is adopted.
- If a filename is specified for line configuration that does not match any name in the configuration file, the access server or router adopts the default specified by the administrator. If one has not been specifically defined, the factory-default VT100 emulation file is used.

Figure 5 illustrates the decision process used by the Cisco IOS software to choose a keymap for a specific TN3270 session. When one of the first four priority checks fails (that is, the name specified does not match any name in the configuration file), the same rules listed for the terminal emulation file apply.

**Figure 5** Decision Diagram for Cisco IOS Software Keymap Selection Process



## Using the Default Terminal Emulation File to Connect

By default, an ASCII terminal and keyboard connected to the Cisco device emulate a Digital VT100 terminal type.

To connect to an IBM host, enter the **tn3270** command from EXEC mode. This command will make the connection using the terminal emulation file selected using the startup sequence priorities outlined in “Startup Sequence Priorities” earlier in this section.

Refer to the “Configuring TN3270 Connections” section later in this document for more information about making connections.

## Copying a Sample Terminal Emulation File

If the default file does not work for your terminal and keyboard type or the host that you connect to, you might be able to find a usable file from the growing list of sample terminal emulation files created by Cisco engineers and customers. You can obtain the TN3270 examples from Cisco.com. Numerous emulation files are listed in the examples, which allow various terminal types to emulate an IBM 3270-type terminal.

To obtain these sample configuration files, perform the following steps:

- Step 1** Obtain a sample configuration file from the following URL. The *TN3270 Keymap Examples* document appears. Note that this URL is subject to change without notice.

<http://www.cisco.com/warp/public/494/1.html>

```
TN3270 Keymap Examples
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! TN3270 examples file
! For use with the TN3270 on the cisco terminal server
! If you have requests for additions, contact tac@cisco.com
! If you have contributions, send them to remaker@cisco.com
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! Example of a ttycap for a televideo 925
! Taken from standard TTYCAP from BSD Unix
!
ttycap televideo \
v8|vi|tvi925|925|televideo model 925:\
      :hs:am:bs:co#80:li#24:cm=\E=%+ %+ :cl=\E*:cd=\Ey:ce=\Et:\

:a1=\EE:dl=\ER:im=:ei=:ic=\EQ:dc=\EW:mr=\EG4:mk=\EG1:md=\EG4:me=\EG0:\
      :ho=^^:nd=^L:bt=\EI:pt:so=\EG4:se=\EG0:sg#1:us=\EG8:ue=\EG0:ug#1:\
      :up=^K:do=^V:kb=^H:ku=^K:kd=^V:kl=^H:kr=^L:kh=^^:ma=^V^J^L :\  

      :k1=^A@\:r:k2=^AA\:r:k3=^AB\:r:k4=^AC\:r:k5=^AD\:r:k6=^AE\:r:k7=^AF\:r\  

      :k8=^AG\:r:k9=^AH\:r:k0=^AI\:r:ko=ic,dc,al,d1,c1,ce,cd,bt:\
      :ts=\Ef:fs=\Eg:ds=\Eh:sr=\Ej:xn:ti=\EG0:to=\EG0:\
      :is=\E1\E"^\M\E3^M          \E1          \E1          \E1          \E1
\E1          \E1          \E1          \E1          \E1^M
!
! Example of a keymap for a 925
! Borrowed from MAP3270 of the BSD TN3270
!
...
```

- Step 2** Use a text editor or word processing application to copy the sample terminal emulation file into the configuration file.
- Step 3** Load the configuration file onto the host or network. (Refer to the chapter “Loading System Images and Configuration Files” in the *Cisco IOS Configuration Fundamentals Configuration Guide*, for information on loading configuration files.)

This procedure adds new terminal emulation capability to the configuration file. Each time the system is started up, or booted, the settings in the file will be used as the default for terminal emulation.

# TN3270 Configuration Task List

To configure TN3270, perform the tasks in the following sections:

- [Configuring TN3270 Connections](#) (Required for Service)
- [Mapping TN3270 Characters](#) (As Required)
- [Starting TN3270 Sessions](#) (Required for Making Connections)

The section “[TN3270 Configuration and Connection Examples](#)” later in this chapter provides examples of making TN3270 connections.

## Configuring TN3270 Connections

The tasks in this section indicate how to create TTYcap and keymap files, and configure your lines for a TN3270 connection.

To create a TTYcap and keymap file, use the following commands in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>ttycap</b> <i>ttycap-name</i> <i>termcap-entry</i>	Creates a custom terminal emulation file, or TTYcap.
Step 2	Router(config)# <b>keymap</b> <i>keymap-name</i> <i>keymap-entry</i>	Creates a custom keyboard emulation file, or keymap.

To configure your line for the TN3270 connection, use the following commands in line configuration mode:

	Command	Purpose
Step 1	Router(config-line)# <b>terminal-type</b> <i>terminal-name</i>	Specifies the type of terminal connected to the line.
Step 2	Router(config-line)# <b>keymap-type</b> <i>keymap-name</i>	Specifies the keyboard map for a terminal connected to the line.

To customize the TN3270 connection environment, use the following commands in global configuration mode. (These tasks are optional).

	Command	Purpose
Step 3	Router(config)# <b>tn3270 datastream</b> { <b>extended</b>   <b>normal</b> }	Enables TN3270 extended features.
Step 4	Router(config)# <b>tn3270 null-processing</b> [ <b>3270</b>   <b>7171</b> ]	Enables null processing.
Step 5	Router(config)# <b>tn3270 reset-required</b>	Specifies a reset whenever a 3278-x terminal keyboard locks up.

To use a custom emulation file, you must load the emulation settings into the system configuration file. This step establishes the settings in the file as the terminal and keyboard defaults and provides several ways in which the emulation settings can be used within the system, as follows:

- You can provide default settings for all terminals in the network or terminals on a specific host.

- You can set up your system to boot, or load, a specific configuration file using configuration commands described in the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.2.
- You can temporarily override default settings using terminal EXEC commands.
- Load in the files by using the local **terminal terminal-type** and **terminal keyboard-type** EXEC commands.
- You can configure line-specific emulation types for terminal negotiations with a remote host.

If you intend to use an alternate TTYcap and keymap, you must assign the following two characteristics:

- Terminal type
- Keymap type

The terminal and keymap type information is used by the Cisco IOS software when negotiating connections with hosts. Use the **terminal-type** and **keymap-type** line configuration commands to assign TTYcap and keymap line characters. You must assign the terminal and keyboard type to the line if you intend to use alternate TTYcap and keymap files.

Use the **tn3270 datastream** command to cause an “-E” to be appended to the terminal type string sent to the IBM host. This command allows you to use the extended TN3270 features.

If a user enters data, uses an arrow key to move the cursor to the right on the screen, and then enters more data, the intervening spaces are filled in with nulls. To specify how nulls are handled, enter the **tn3270 null-processing** command either with the argument **3270**, where nulls are compressed out of the string (as on a real 3278-x terminal), or use the **7171** argument, where nulls are converted to spaces as on a 7171 controller.

On a 3278-x terminal, the keyboard is locked and further input is not permitted after an input error (due to field overflow, invalid entry, and so on), until the user presses the RESET key. Most TN3270 implementations leave the keyboard unlocked and remove any error message on the next key input after the error. Use the **tn3270 reset-required** command to enable a reset in these situations.

## Mapping TN3270 Characters

To control the mapping of EBCDIC and ASCII characters, use the following commands in the modes indicated, as needed:

Command	Purpose
Router(config)# <b>tn3270 character-map</b> <i>ebcdic-in-hex</i> <i>ascii-in-hex</i>	In global configuration mode, creates character mappings by configuring a two-way binding between EBCDIC and ASCII characters.
Router(config)# <b>no tn3270 character-map</b> {all   <i>ebcdic-in-hex</i> } [ <i>ascii-in-hex</i> ]	In global configuration mode, resets character mappings to their default settings.
Router> <b>show tn3270 character-map</b> {all   <i>ebcdic-in-hex</i> }	In EXEC mode, displays character mappings.
Router> <b>show tn3270 ascii-hexval</b>	In EXEC mode, displays the hexadecimal value of an ASCII character. <sup>1</sup>

Command	Purpose
Router(config-line)# <b>tn3270 8bit display</b>	In line configuration mode, temporarily configures the Cisco IOS software to use the 8-bit mask.
Router(config-line)# <b>tn3270 8bit transparent-mode</b>	In line configuration mode, temporarily configures the Cisco IOS software to use the 8-bit mask if you use a file-transfer protocol such as Kermit in 8-bit mode.

1. After you enter the **show tn3270 ascii-hexval** command, enter the ASCII character whose hexadecimal value you want to display.

When you create character mappings between extended EBCDIC or extended ASCII characters, you must configure the Cisco IOS software for the correct data character bit length. The default mask used for TN3270 connections is a 7-bit mask. In certain situations, you must use an 8-bit display. When an 8-bit mask has been set by the **data-character-bits {7|8}** line configuration command or the **terminal data-character-bits {7|8}** EXEC command, you can temporarily configure the software to use the 8-bit mask by entering the **tn3270 8bit display** line configuration command.

When you use a file-transfer protocol such as Kermit in 8-bit mode or you use 8-bit graphics, which rely on transparent mode, use the **tn3270 8bit transparent-mode** line configuration command to configure the software for the 8-bit mask.

## Starting TN3270 Sessions

You use TN3270 terminal emulation to connect to an IBM 3278-type host. Your system administrator must configure a default terminal emulation file that permits the terminal to communicate with the host. How to specify alternate terminal emulations is described in the section “[Configuring TN3270 Connections](#)” earlier in this chapter.

Unlike with Telnet and LAT connections, you *must* enter the **tn3270** command to make a connection to an IBM 3278-type host. To start a TN3270 session, use the following command in EXEC mode:

Command	Purpose
Router> <b>tn3270 host</b> [ <i>keyword</i> ]	Begins a TN3270 session. Refer to the description of the <b>tn3270</b> command in the <i>Cisco IOS Terminal Services Command Reference</i> , for a list of supported keywords.

To terminate an active TN3270 session, enter the escape sequence (Ctrl-Shift-6 then x [Ctrl^x] by default) and enter the **disconnect** command at the EXEC prompt. You can also log out of the remote system by issuing the command specific to that system (such as **exit**, **logout**, **quit**, **close**, or **disconnect**). For an example of making TN3270 connections, see the next section, “[TN3270 Configuration and Connection Examples](#).”

## TN3270 Configuration and Connection Examples

This section provides the following examples to help you define custom terminal and keyboard emulation files, and to configure your system to use those files:

- [Custom Terminal Emulation File Example](#)
- [Custom Keyboard Emulation File Example](#)

- [Line Specification for a Custom Emulation Example](#)
- [Character Mapping Examples](#)
- [TN3270 Connection Example](#)

## Custom Terminal Emulation File Example

The following example allows a Televideo 925 terminal to emulate an IBM 3270-type terminal. The file is part of the global **ttycap** command and is included in the system configuration file. Notice that a carriage return (^M) indicates the last character in the file.

```
ttycap ttycap1 \
v8 | vi | tvi925 | 925 | televideo model 925:\
:so=\EG4:se=\EG0:\
:hs:am:bs:co#80:li#24:cm=\E=%+ %+ :cl=\E*:cd=\Ey:ce=\Et:\
:al=\EE:d1=\ER:im=:ei=:ic=\EQ:dc=\EW:\
:ho=^^:nd=\L:bt=\EI:pt:so=\EG4:se=\EG0:sg#1:us=\EG8:ue=\EG0:ug#1:\
:up=^K:do=^V:kb=^H:ku=^K:kd=^V:kl=^H:kr=^L:kh=^^:ma=^V^J^L :\  
:k1=^A@\r:k2=^AA\r:k3=^AB\r:k4=^AC\r:k5=^AD\r:k6=^AE\r:k7=^AF\r:\
:k8=^AG\r:k9=^AH\r:k0=^AI\r:ko=ic,dc,al,d1,c1,ce,cd,bt:\
:md=\E(:me=\E):ti=\E):te=\E(:\  
:ts=\Ef:fs=\Eg:ds=\Eh:sr=\Ej:xn:\
:is=\E1\E"^\M\E3^\M      \E1      \E1      \E1      \E1      \E\  
1      \E1      \E1      \E1      \E1^\M
```

## Custom Keyboard Emulation File Example

The following example allows a keyboard to emulate an asynchronous connection to an IBM 7171 keyboard. The file is part of the **keymap** global configuration command and is included in the system configuration file.

```
keymap ibm7171 \
vt100av | vt100 | vt100nam | pt100 | vt102 | vt125{ \  
enter = '^m';\  
erase = '^?'; reset = '^g'; clear = '^z' | '\EOM';\  
nl = '^j'; tab = '^i'; btab = '^b';\  
left = '\EOD'; right = '\EOC'; up = '\EOA'; down = '\EOB';\  
home = '^h'; delete = '^d'; eof = '^e' | '\E^?'; einp = '^w'; insrt = '\EOn';\  
pfk1 = '\EOP' | '\E1'; pfk2 = '\EOQ' | '\E2'; pfk3 = '\EOR' | '\E3';\  
pfk4 = '\EOW' | '\E4'; pfk5 = '\EOx' | '\E5'; pfk6 = '\EOy' | '\E6';\  
pfk7 = '\EOt' | '\E7'; pfk8 = '\EOu' | '\E8'; pfk9 = '\EOv' | '\E9';\  
pfk10 = '\EOq' | '\E0'; pfk11 = '\EOr' | '\E-';\  
pfk12 = '\EOs' | '\E='; pfk13 = '\EOp\EOP' | '^f13';\  
pfk14 = '\EOp\EOQ' | '^f14'; pfk15 = '\EOp\EOR' | '^f15';\  
pfk16 = '\EOp\EOW' | '^f16'; pfk17 = '\EOp\EOx' | '^f17';\  
pfk18 = '\EOp\EOy' | '^f18'; pfk19 = '\EOp\EOt' | '^f19';\  
pfk20 = '\EOp\EOu' | '^f20'; pfk21 = '\EOp\EOv' | '^f21';\  
pfk22 = '\EOp\EOq' | '^f22'; pfk23 = '\EOp\EOr' | '^f23';\  
pfk24 = '\EOp\EOs' | '^f24';\  
pa1 = '^p1' | '\EOS';\  
pa2 = '^p2' | '\EOM';\  
pa3 = '^p3' | '\EOL';\  
}
```

## Line Specification for a Custom Emulation Example

The following example sets up a line with specific terminal and keyboard characteristics that are used during negotiation with a host upon connection. The line configuration commands in the example must follow the global **ttycap** and **keymap** global configuration commands containing the emulation settings to be used.

```
line 3
  terminal-type ttycap1
  keymap-type ibm7171
```

## Character Mapping Examples

The following example shows the configuration of the EBCDIC and ASCII character mappings listed in [Table 1](#):

```
tn3270 character-map 0x81 0x78
tn3270 character-map 0x82 0x79
tn3270 character-map 0x83 0x7A
```

**Table 1** Sample EBCDIC and ASCII Character Mapping

EBCDIC	ASCII
a	x
b	y
c	z

The following example displays all nonstandard character mappings:

```
Router# show tn3270 character-map all

EBCDIC 0x81 <=> 0x78 ASCII
EBCDIC 0x82 <=> 0x79 ASCII
EBCDIC 0x83 <=> 0x7A ASCII
```

The following example shows the standard key mapping for the letters d and c:

```
Router# show tn3270 character-map 83

EBCDIC 0x83 <=> 0x63 ASCII = `c`
EBCDIC 0x84 <=> 0x64 ASCII = `d`
```

The following example unmaps a specific key, first with the optional *ascii-in-hex* argument and then without the argument:

```
Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# no tn3270 character-map 0x80 0x78
Router(config)# ^Z

Router# show tn3270 character-map all

EBCDIC 0x82 <=> 0x79 ASCII
EBCDIC 0x83 <=> 0x7A ASCII

Router# configure terminal
```



```

Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# no tn3270 character-map 0x82
Router(config)# ^Z
Router# show t3270 character-map all

```

```
EBCDIC 0x82 <=> 0x79 ASCII
```

The following example displays character mappings, then removes all mappings with the **all** keyword:

```
Router# show tn3270 character-map all
```

```
EBCDIC 0x81 <=> 0x78 ASCII
EBCDIC 0x82 <=> 0x79 ASCII
EBCDIC 0x83 <=> 0x7A ASCII
```

```
Router# configure terminal
```

```

Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# no tn3270 character-map all
Router(config)# ^Z

```

```
Router# show tn3270 character-map all
```

## TN3270 Connection Example

The following example establishes a terminal session with an IBM TN3270 host named *finance* and specifies *vt100* as the terminal type:

```
Router> tn3270 finance /terminal-type vt100
```

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

## TN3270 Menu Example

The following example shows the use of the **/terminal-type** *type* keyword and argument combination when using *tn3270* with menus:

```

menu router1 text 1 Connect from client
  menu router1 command 1 tn3270 router1.com /term h19
  menu router1 text 2 Connect from VT-100
  menu router1 command 2 tn3270 router1.com /term vt100
  menu router1 text 3 Connect from PC running Procomm
  menu router1 tn3270 router1.com /term vt100-pc

```

## Configuring XRemote

The X Window System, also called X, is a network-based graphics window system originally developed for workstations running UNIX. Cisco has developed an XRemote application that allows the XRemote capabilities of X terminals to run on an access server or router.

Previous window systems for terminals were *kernel-based* and therefore were closely linked to the operating system running on the workstation itself. They typically only ran on discrete systems, such as a single workstation. The X Window System is not part of any operating system, but instead, is composed of application programs. Thus, the X Window System enables flexible, graphics-based network computing across a wide range of operating systems and hardware platforms.

## X and the Client/Server Model

The underlying architecture of the X Window System is based on a *client/server* model. The system is split into two parts: *clients* and *display servers*. Clients are application programs that perform specific tasks, and display servers provide specific display capabilities and track user input. These two parts can reside on the same computer or can be separated over a network. In an X terminal environment, such as in NCD terminal implementations, the display server resides on the display station and the client resides on a host computer.

Because the X Windows System employs this client/server partitioning and is independent of both the hardware and operating environment, X terminal users can access different types of computers to simultaneously access several applications and resources in a multivendor environment. A user at an X terminal can concurrently run and display a calendar program on a VAX, a spreadsheet program on a PC, and a compiler on a workstation.

## XRemote Overview

**Note**

---

Beginning with Cisco IOS XE Gibraltar 16.11.1, this protocol is no longer supported.

---

XRemote is a protocol developed specifically to optimize support for the X Window System over a serial communications link. Its compression and decompression algorithms are designed to handle bit-mapped displays and windowing systems.

There are two basic parts to XRemote:

- Server-side helper process
- Client-side helper process

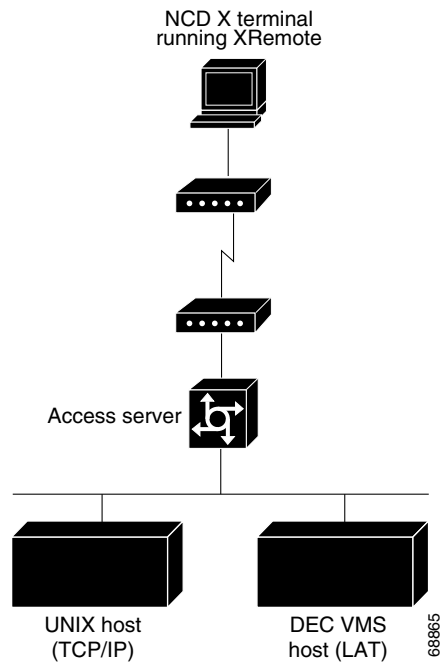
These two helper processes communicate with each other using the XRemote protocol. The client-side helper communicates with X clients using the standard X protocol. The server-side helper communicates with the server using the standard X Window System. The server-side helper might operate as part of the X server or it might be external and accessed across the network; for example, the server-side helper can operate in an access server or router at your house or work site. If the server-side helper is in the X terminal, it must have XRemote programmable read-only memory (PROM) installed.

XRemote enables a user of a display station to run the X Window System via 9600-baud (and faster) modem connections with performance that is superior to using conventional serial protocols, such as Serial Line Internet Protocol (SLIP). An X display station must either implement XRemote or be connected to a network configuration that includes an access server or router.

## Connection Capability

The Cisco implementation of XRemote is fully compatible with the NCD XRemote protocol. [Figure 6](#) illustrates an XRemote connection between an X terminal and an access server. In [Figure 6](#), the server-side helper runs on the X terminal, and the client-side helper runs on the access server.

**Figure 6** XRemote Session from an X Display Server Running XRemote



## Remote Access to Fonts

Remote access to fonts is provided in three ways:

- Using the industry-standard protocol for transporting X traffic over TCP/IP networks
- Using the Digital protocol for transporting X traffic over LAT networks
- Using the Internet standard TFTP for TCP/IP networks

A single XRemote user can use any combination of TCP/IP and LAT client connections and any combination of TFTP and LAT font access.

## XRemote Configuration Task List

To configure XRemote, perform the tasks described in the following sections:

- [Configuring XRemote](#) (Required for Service)
- [Selecting Fonts for X Terminal Applications](#) (Optional)
- [Making XRemote Connections](#) (Required for Making Connections)

The section “[Monitoring XRemote Connections](#)” provides tips on maintaining XRemote connections.

## Configuring XRemote

To allow host connections using the XRemote feature from NCD and the access server or router, use the following commands. Before starting the following tasks, verify that a modem is externally or internally connected with your access server or router. Unless specified otherwise, all commands in this task table are entered in global configuration mode.

	Command	Purpose <sup>1</sup>
Step 1	Router(config)# <b>xremote tftp host</b> <i>hostname</i>	Defines a specific TFTP font server as the source for fonts.
Step 2	Router(config)# <b>xremote tftp buffersize</b> <i>buffersize</i>	Sets the buffer size used for loading font files.
Step 3	Router(config)# <b>xremote tftp retries</b> <i>retries</i>	Increases the number of times that the font loader tries to load the fonts. <sup>2</sup>
Step 4	Router> <b>show xremote</b>	(Optional) In EXEC mode, displays current XRemote connections and monitors traffic.
Step 5	Router> <b>show xremote line</b> <i>number</i>	(Optional) In EXEC mode, displays XRemote traffic and line statistics.

1. The X Server for the X terminal and the network and serial parameters for the X terminal must be configured as described in the publications for the specific X terminal you are using. In general, the X terminal configuration determines the mode of operation for the terminal, the source of font information, and the source of remote configuration information (when applicable).
2. This feature is particularly useful when the font servers are known to be heavily loaded.

In general, you can use any modem that provides acceptable performance for your application. The following guidelines apply to an XRemote operation using a modem (see the user manual for your modem for specific connection procedures):

- Attach cables and set up your modem for use with XRemote (access over asynchronous lines only), or cable the X terminal directly to the access server or router.
- Disable any error correction and compression features of the modem. Because XRemote implements its own compression and error correction, the compression and error correction from the modem actually impair performance.
- If you must use a flow control mechanism, hardware flow control (such as RTS/CTS or DTR/DSR) is recommended. Software flow control (such as XON/XOFF) is discouraged.
- The modem should incur minimal delays in round-trip transmissions, even when transmitting small packets, and transmissions should be transparent to the data stream.
- The modem should provide true full-duplex transmission at 9600 baud or faster. Half-duplex modems are not suitable for use with XRemote.

Refer to *Cisco IOS Dial Technologies Configuration Guide*, for more information about configuring modems.

When the X terminal requests that a font file be loaded, the Cisco IOS software must first load the font file into an internal buffer before passing it to the X terminal. The default value for this buffer is 70000 bytes, which is adequate for most font files, but the size can be increased as necessary for nonstandard font files using the **xremote tftp buffersize** global configuration command. This task can be performed for both TFTP and LAT font access.

## Selecting Fonts for X Terminal Applications

The NCD terminal contains a small set of built-in fonts in local ROM. You should use these fonts because loading fonts over a serial line can increase application startup time. The default for an NCD terminal is to use built-in fonts, unless you log in using DECwindows over LAT. When using DECwindows over LAT, the standard DECwindows fonts are used automatically.

To select fonts, perform the tasks described in the following sections:

- [Accessing Nonresident Fonts Using TFTP](#)
- [Selecting DECwindows Fonts](#)

### Accessing Nonresident Fonts Using TFTP

When an X terminal application requests a font that is not stored in ROM for the terminal, the X terminal makes a request for a font file from the access server or router. The Cisco IOS software uses the TFTP to load the font from the font server, and then passes the font to the X terminal using the XRemote protocol. Loading fonts from the access server or router to the X terminal can take 30 to 45 seconds, depending on the size of the font file.

An X server can display only the fonts it finds in the directories in its font path. The default font path for the X server includes only the built-in fonts. To access fonts stored on a host, you must add the font directories from the host to the font path of the X server, which is done using the UNIX command **xset** with the **fp+** argument to add fonts to the end of the font path of the server.

For example, to allow your display station to access the 100 dots-per-inch (dpi) fonts found in the standard font directory, enter the following command at the host system prompt:

```
host_prompt% xset fp+ /usr/lib/x11/ncd/fonts/100dpi
```

For more information, see the *NCDware XRemote User's Manual*.

### Selecting DECwindows Fonts

Downloading of fonts occurs automatically when you initiate a remote DECwindows login session using the **xremote lat EXEC** command. Using the **xremote lat EXEC** command instead of relying on TFTP to download the fonts, the fonts are read in via the LAT protocol.

If you want to use DECwindows fonts while running standard X applications on a UNIX host, you need to use the UNIX **xset** command or an application that sends an XSetFontPath request to set a font path. You might want to use the UNIX **xset** command if you are primarily a TCP/IP user, but also run some DECwindows applications.

Enter the **xset** command, or launch the application that sends an XSetFontPath request, to set the following path:

```
/LAT/SERVICE
```

In this path, SERVICE is a LAT service name with DECwindows support; case is not significant.

When the Cisco IOS software sees a request for font files in that directory, it uses LAT instead of TFTP to access the specified service.

## Making XRemote Connections

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

- [Connecting Through Automatic Session Startup with an XDMCP Server](#)
- [Connecting Through Automatic Session Startup with a DECwindows Login via LAT](#)
- [Connecting Through Manual XRemote Session Startup](#)
- [Establishing XRemote Sessions Between Servers](#)
- [Exiting XRemote Sessions](#)

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

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

For an example of making an XRemote connection, see the “[XRemote Configuration and Connection Examples](#)” section later in this chapter.

### Connecting Through Automatic Session Startup with an XDMCP Server

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

Command	Purpose
Router> <code>xremote xdm [hostname]</code>	Creates a connection with XRemote and an XDMCP server.

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

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

### Connecting Through Automatic Session Startup with a DECwindows Login via LAT

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

Command	Purpose
Router> <code>xremote lat service</code>	Creates a connection with XRemote and DECwindows over LAT.

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

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

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



**Note**

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

## Connecting Through Manual XRemote Session Startup

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

- [Enabling XRemote Manually](#) (Required for Manual Sessions)
- [Connecting to the Remote Host Computer](#) (Required for Manual Sessions)
- [Setting the Location of the X Display](#) (Required for Manual Sessions)
- [Starting Client Applications](#) (Required for Manual Sessions)
- [Returning to the EXEC Prompt](#) (Required for Manual Sessions)
- [Reenabling XRemote Manually](#) (Required for Manual Sessions)

## Enabling XRemote Manually

To prepare the XRemote server for manual startup, use the following command in EXEC mode:

Command	Purpose
Router> <b>xremote</b>	Prepares the XRemote server for manual startup.

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



**Note**

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

## Connecting to the Remote Host Computer

To connect to a host, use one of the following commands in EXEC mode:

Command	Purpose
Router> <b>telnet</b> OR Router> <b>lat</b> OR Router> <b>rlogin</b>	Prepares the server for XRemote manual startup.

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

## Setting the Location of the X Display



### Note

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

Once you are logged in to the remote host computer, inform the host computer of your X display location that the server provided when you enabled XRemote manually. For most versions of the UNIX operating system, the X display location is set by using the **setenv** command to set the Display environment variable. Refer to the online X(1) manual page available from UNIX for more information.

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



### Note

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

## Starting Client Applications

When you have set the location of the Xdisplay, you can start your client applications for your host operating system, as specified in the documentation for the client applications.

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

## Returning to the EXEC Prompt

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

If you cannot log out of the host computer and keep your clients running, return to the EXEC prompt for the access server using the escape sequence (Ctrl-Shift-6 then x [Ctrl^x] by default).



## Reenabling XRemote Manually

To begin a manual remote session again, see the “[Enabling XRemote Manually](#)” section earlier in this chapter. If the X clients connected successfully, the session is put into XRemote mode, and the clients complete their startup.

If no clients are found, you see the following message: “No X clients waiting - check that your display is darkstar:2018”

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

## Establishing XRemote Sessions Between Servers

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

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

**Note**

---

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

---

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

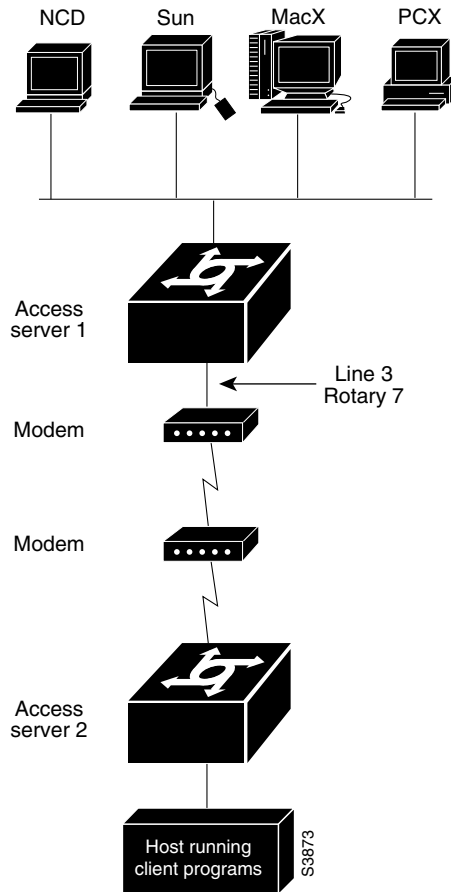
Depending upon the connection configuration, use one of the following connection methods:

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

For information about how to configure individual lines and rotary connections, refer to *Cisco IOS Dial Technologies Configuration Guide*.

[Figure 7](#) illustrates a configuration in which a display server is not running XRemote. In this configuration, the server-side XRemote helper is running on the access server named Access Server 1, and the client-side XRemote helper is running on the access server named Access Server 2.

**Figure 7** XRemote Session Between Servers



## Exiting XRemote Sessions

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

## Monitoring XRemote Connections

To list XRemote connections and monitor XRemote traffic through the router, use the following commands in EXEC mode as needed:

Command	Purpose
Router> <b>show xremote</b>	Lists XRemote connections and monitors XRemote traffic through the router or access server.
Router> <b>show xremote line number</b>	Lists XRemote connections and monitors XRemote traffic for specific lines on an XRemote server.

# XRemote Configuration and Connection Examples

These examples are provided to help you understand how to make XRemote connections:

- [Standard XRemote Configuration Example](#)
- [Connecting Through Automatic Session Startup with XDMCP Server Example](#)
- [Connecting Through Automatic Session Startup with DECwindows Login via LAT Example](#)
- [Enabling XRemote Manually Example](#)
- [Connecting an X Display Terminal Example](#)
- [Making XRemote Connections Between Servers Example](#)

## Standard XRemote Configuration Example

The following example shows how to specify IBM-1 as the host name of the TFTP font server, how to specify 7 retry attempts at accessing the server, and how to reduce the buffer size to 20,000 bytes:

```
xremote tftp host IBM-1
xremote tftp retries 7
xremote tftp buffersize 20000
```

## Connecting Through Automatic Session Startup with XDMCP Server Example

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

```
Router> xremote xdm star
```

## Connecting Through Automatic Session Startup with DECwindows Login via LAT Example

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

```
Router> xremote lat WHIRL
```

## Enabling XRemote Manually Example

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

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

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

If no clients are found, you see the following message: “No X clients waiting - check that your display is darkstar:2006”

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

## Connecting an X Display Terminal Example

To make a connection from an X display terminal through a server to a host running client programs, perform the following steps:

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

```
Router> xremote
```

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

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

**Step 3** Connect to the client:

```
Router> telnet eureka
Trying EUREKA.NOWHERE.COM (172.16.1.55)... Open

SunOS UNIX (eureka)
```

**Step 4** Log in at the prompt:

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

**Step 5** At the client prompt, enter the display name from Step 2 in this procedure and the **xterm** command:

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

**Step 6** Disconnect from the client:

```
eureka% logout

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

**Step 7** Begin the XRemote session:

```
Router> xremote
Entering XRemote
```

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

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

## Making XRemote Connections Between Servers Example

This section describes two ways to make XRemote connections between servers.

The following process explains how an XRemote connection is established for a configuration such as the one shown in [Figure 7](#) in the section “[Establishing XRemote Sessions Between Servers](#)” earlier in this chapter. This procedure assumes that the administrator has set the display environment variable to identify and match the X display terminal of the user.

From the PCX, MacX, or UNIX machine in [Figure 7](#), the user connects to port 9003 on the access server named Access Server 1. If your administrator has configured a rotary number 7, the user connects to port 10007. For more information about rotary groups, refer to *Cisco IOS Dial Technologies Configuration Guide*.

Following is a summary of the connection process:

1. Access Server 1 connects the user to a modem.
2. The modem calls Access Server 2.
3. The user enters the **xremote** command at the Access Server 2 prompt.
4. The user connects to the remote host from Access Server 2 using the **telnet** command.
5. The user starts the X client program that runs on the remote host and displays on the X display server (PCX, MacX, or UNIX host).
6. The user escapes from the remote host back to Access Server 2, or logs out if clients were run in the background, and enters the **xremote** command again at the Access Server 2 prompt.

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

---

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

```
space% telnet golden-road 9016
Trying 172.31.7.84 ...
Connected to golden-road.cisco.com.
Escape character is '^]'.
```

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

```
User Access Verification

Password: <password>
Password OK

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

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

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

```
atdt 13125554141
DIALING
RING
CONNECT 14400

User Access Verification
Username: deal
Password:
Welcome to the cisco dial-up access server.
```

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

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

**Step 5** Connect to the client:

```
Router> telnet sparks
Trying SPARKS.NOWHERE.COM (173.19.1.55)... Open

SunOS UNIX (sparks)

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

**Step 6** At the client prompt, enter the display name from [Step 4](#) and the **xterm** command:

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

**Step 7** Disconnect from the client:

```
sparks% logout

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

**Step 8** Begin the XRemote session.

```
Router> xremote
Entering XRemote
```

When the connection is closed by the foreign host, the Xterm window appears on the local workstation screen:

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

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 20018 Cisco Systems, Inc. All rights reserved.