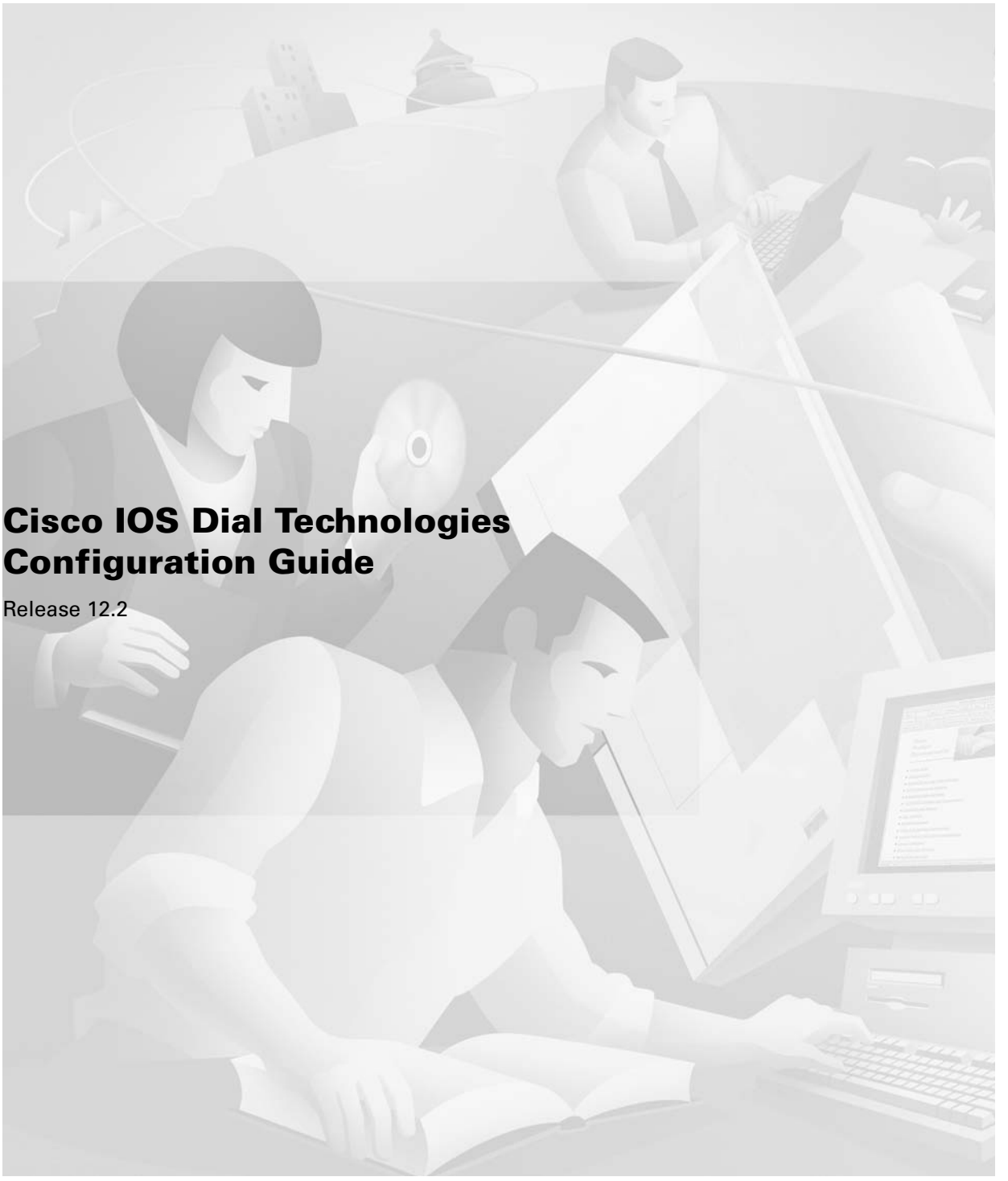


Cisco IOS Dial Technologies Configuration Guide

Release 12.2

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: DOC-7812090=
Text Part Number: 78-12090-02

CONTENTS

DIAL INTERFACES, CONTROLLERS, AND LINES

ISDN CONFIGURATION

SIGNALING CONFIGURATION

VIRTUAL TEMPLATES, PROFILES, AND NETWORKS

PPP CONFIGURATION

CALLBACK AND BANDWIDTH ALLOCATION CONFIGURATION

About Cisco IOS Software Documentation

This chapter discusses the objectives, audience, organization, and conventions of Cisco IOS software documentation. It also provides sources for obtaining documentation from Cisco Systems.

Documentation Objectives

Cisco IOS software documentation describes the tasks and commands necessary to configure and maintain Cisco networking devices.

Audience

The Cisco IOS software documentation set is intended primarily for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the tasks, the relationship between tasks, or the Cisco IOS software commands necessary to perform particular tasks. The Cisco IOS software documentation set is also intended for those users experienced with Cisco IOS software who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS software release.

Documentation Organization

The Cisco IOS software documentation set consists of documentation modules and master indexes. In addition to the main documentation set, there are supporting documents and resources.

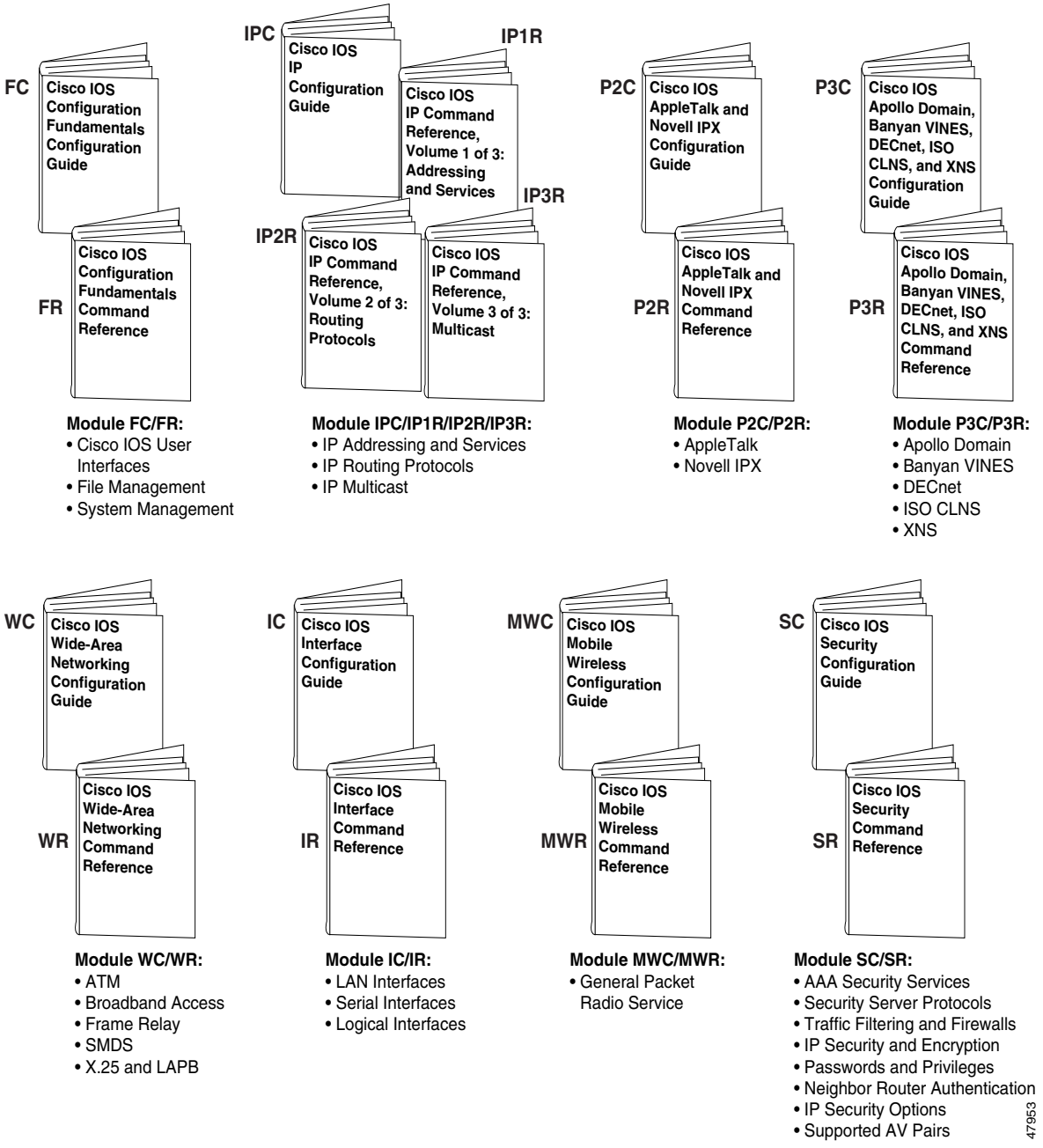
Documentation Modules

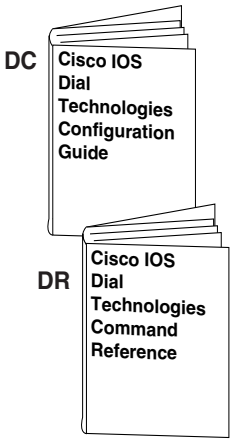
The Cisco IOS documentation modules consist of configuration guides and corresponding command reference publications. Chapters in a configuration guide describe protocols, configuration tasks, and Cisco IOS software functionality and contain comprehensive configuration examples. Chapters in a command reference publication provide complete Cisco IOS command syntax information. Use each configuration guide in conjunction with its corresponding command reference publication.

Figure 1 shows the Cisco IOS software documentation modules.

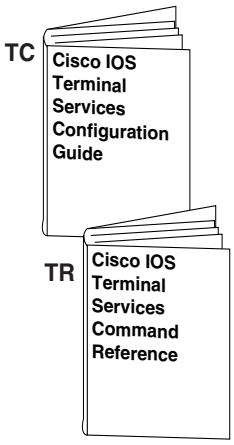
Note The abbreviations (for example, FC and FR) next to the book icons are page designators, which are defined in a key in the index of each document to help you with navigation. The bullets under each module list the major technology areas discussed in the corresponding books.

Figure 1 Cisco IOS Software Documentation Modules

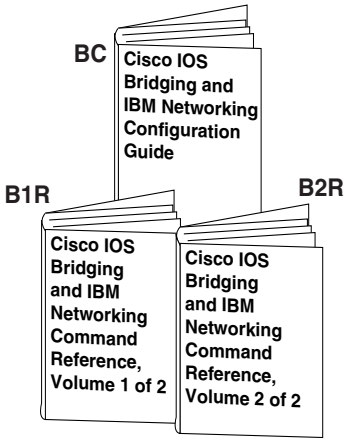




Module DC/DR:
- Preparing for Dial Access
- Modem and Dial Shelf Configuration and Management
- ISDN Configuration
- Signalling Configuration
- Dial-on-Demand Routing Configuration
- Dial-Backup Configuration
- Dial-Related Addressing Services
- Virtual Templates, Profiles, and Networks
- PPP Configuration
- Callback and Bandwidth Allocation Configuration
- Dial Access Specialized Features
- Dial Access Scenarios

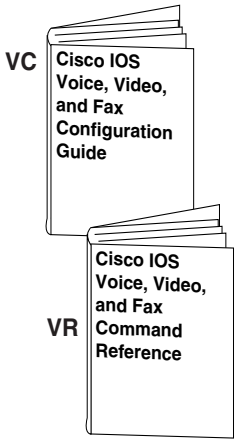


Module TC/TR:
- ARA
- LAT
- NASI
- Telnet
- TN3270
- XRemote
- X.28 PAD
- Protocol Translation

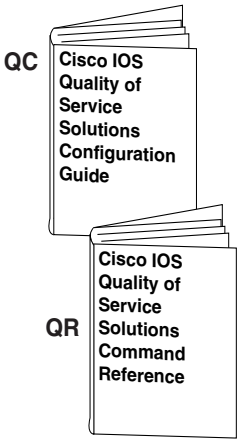


Module BC/B1R:
- Transparent Bridging
- SRB
- Token Ring Inter-Switch Link
- Token Ring Route Switch Module
- RSRB
- DLSw+
- Serial Tunnel and Block Serial Tunnel
- LLC2 and SDLC
- IBM Network Media Translation
- SNA Frame Relay Access
- NCIA Client/Server
- Airline Product Set

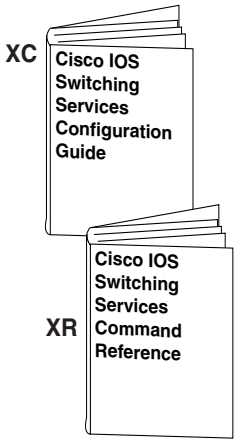
Module BC/B2R:
- DSPU and SNA Service Point
- SNA Switching Services
- Cisco Transaction Connection
- Cisco Mainframe Channel Connection
- CLAW and TCP/IP Offload
- CSNA, CMPC, and CMPC+
- TN3270 Server



Module VC/VR:
- Voice over IP
- Call Control Signalling
- Voice over Frame Relay
- Voice over ATM
- Telephony Applications
- Trunk Management
- Fax, Video, and Modem Support



Module QC/QR:
- Packet Classification
- Congestion Management
- Congestion Avoidance
- Policing and Shaping
- Signalling
- Link Efficiency Mechanisms



Module XC/XR:
- Cisco IOS Switching Paths
- NetFlow Switching
- Multiprotocol Label Switching
- Multilayer Switching
- Multicast Distributed Switching
- Virtual LANs
- LAN Emulation

47954

Master Indexes

Two master indexes provide indexing information for the Cisco IOS software documentation set: an index for the configuration guides and an index for the command references. Individual books also contain a book-specific index.

The master indexes provide a quick way for you to find a command when you know the command name but not which module contains the command. When you use the online master indexes, you can click the page number for an index entry and go to that page in the online document.

Supporting Documents and Resources

The following documents and resources support the Cisco IOS software documentation set:

- *Cisco IOS Command Summary* (two volumes)—This publication explains the function and syntax of the Cisco IOS software commands. For more information about defaults and usage guidelines, refer to the Cisco IOS command reference publications.
- *Cisco IOS System Error Messages*—This publication lists and describes Cisco IOS system error messages. Not all system error messages indicate problems with your system. Some are purely informational, and others may help diagnose problems with communications lines, internal hardware, or the system software.
- *Cisco IOS Debug Command Reference*—This publication contains an alphabetical listing of the **debug** commands and their descriptions. Documentation for each command includes a brief description of its use, command syntax, usage guidelines, and sample output.
- *Dictionary of Internetworking Terms and Acronyms*—This Cisco publication compiles and defines the terms and acronyms used in the internetworking industry.
- New feature documentation—The Cisco IOS software documentation set documents the mainline release of Cisco IOS software (for example, Cisco IOS Release 12.2). New software features are introduced in early deployment releases (for example, the Cisco IOS "T" release train for 12.2, 12.2(x)T). Documentation for these new features can be found in standalone documents called "feature modules." Feature module documentation describes new Cisco IOS software and hardware networking functionality and is available on Cisco.com and the Documentation CD-ROM.
- Release notes—This documentation describes system requirements, provides information about new and changed features, and includes other useful information about specific software releases. See the section "Using Software Release Notes" in the chapter "Using Cisco IOS Software" for more information.
- Caveats documentation—This documentation provides information about Cisco IOS software defects in specific software releases.
- RFCs—RFCs are standards documents maintained by the Internet Engineering Task Force (IETF). Cisco IOS software documentation references supported RFCs when applicable. The full text of referenced RFCs may be obtained on the World Wide Web at http://www.rfc-editor.org/.
- MIBs—MIBs are used for network monitoring. For lists of supported MIBs by platform and release, and to download MIB files, see the Cisco MIB website on Cisco.com at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

New and Changed Information

For Cisco IOS Release 12.2, two previous Release 12.1 guides, *Cisco IOS Dial Services Configuration Guide: Terminal Services* and *Cisco IOS Dial Services Configuration Guide: Network Services*, have been renamed and reorganized into a single book: *Cisco IOS Dial Technologies Configuration Guide*. See Figure 1 for a list of the contents.

For Cisco IOS Release 12.2, the Release 12.1 *Cisco IOS Dial Services Command Reference* has been renamed *Cisco IOS Dial Technologies Command Reference*.

The *Cisco IOS Terminal Services Configuration Guide* and *Cisco IOS Terminal Services Command Reference* were extracted from the 12.1 release of the *Cisco IOS Dial Services Configuration Guide: Terminal Services* and *Cisco IOS Dial Services Command Reference,* and placed in separate books not included in this set.

Document Conventions

Within Cisco IOS software documentation, the term *router* is generally used to refer to a variety of Cisco products (for example, routers, access servers, and switches). Routers, access servers, and other networking devices that support Cisco IOS software are shown interchangeably within examples. These products are used only for illustrative purposes; that is, an example that shows one product does not necessarily indicate that other products are not supported.

The Cisco IOS documentation set uses the following conventions:

Convention	Description
^ or Ctrl	The ^ and Ctrl symbols represent the Control key. For example, the key combination ^D or Ctrl-D means hold down the Control key while you press the D key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

Command syntax descriptions use the following conventions:

Convention	Description
boldface	Boldface text indicates commands and keywords that you enter literally as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example:

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`boldface screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. (Exclamation points are also displayed by the Cisco IOS software for certain processes.)
[ ]	Square brackets enclose default responses to system prompts.

The following conventions are used to attract the attention of the reader:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials not contained in this manual.

Timesaver Means the *described action saves time*. You can save time by performing the action described in the paragraph.

Obtaining Documentation

The following sections provide sources for obtaining documentation from Cisco Systems.

World Wide Web

The most current Cisco documentation is available on the World Wide Web at the following website:

http://www.cisco.com

Translated documentation is available at the following website:

http://www.cisco.com/public/countries_languages.html

Documentation CD-ROM

Cisco documentation and additional literature are available in a CD-ROM package, which ships with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual subscription.

Ordering Documentation

Cisco documentation can be ordered in the following ways:

- Registered Cisco Direct Customers can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/cgi-bin/order/order_root.pl

- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

  http://www.cisco.com/go/subscription

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco corporate headquarters (California, USA) at 408 526-7208 or, in North America, by calling 800 553-NETS(6387).

Documentation Feedback

If you are reading Cisco product documentation on the World Wide Web, you can submit technical comments electronically. Click **Feedback** in the toolbar and select **Documentation**. After you complete the form, click **Submit** to send it to Cisco.

You can e-mail your comments to bug-doc@cisco.com.

To submit your comments by mail, use the response card behind the front cover of your document, or write to the following address:

Cisco Systems, Inc.
Document Resource Connection
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com as a starting point for all technical assistance. Customers and partners can obtain documentation, troubleshooting tips, and sample configurations from online tools. For Cisco.com registered users, additional troubleshooting tools are available from the TAC website.

Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information and resources at anytime, from anywhere in the world. This highly integrated Internet application is a powerful, easy-to-use tool for doing business with Cisco.

Cisco.com provides a broad range of features and services to help customers and partners streamline business processes and improve productivity. Through Cisco.com, you can find information about Cisco and our networking solutions, services, and programs. In addition, you can resolve technical issues with online technical support, download and test software packages, and order Cisco learning materials and merchandise. Valuable online skill assessment, training, and certification programs are also available.

Customers and partners can self-register on Cisco.com to obtain additional personalized information and services. Registered users can order products, check on the status of an order, access technical support, and view benefits specific to their relationships with Cisco.

To access Cisco.com, go to the following website:

http://www.cisco.com

Technical Assistance Center

The Cisco TAC website is available to all customers who need technical assistance with a Cisco product or technology that is under warranty or covered by a maintenance contract.

Contacting TAC by Using the Cisco TAC Website

If you have a priority level 3 (P3) or priority level 4 (P4) problem, contact TAC by going to the TAC website:

http://www.cisco.com/tac

P3 and P4 level problems are defined as follows:

- P3—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- P4—You need information or assistance on Cisco product capabilities, product installation, or basic product configuration.

In each of the above cases, use the Cisco TAC website to quickly find answers to your questions.

To register for Cisco.com, go to the following website:

http://www.cisco.com/register/

If you cannot resolve your technical issue by using the TAC online resources, Cisco.com registered users can open a case online by using the TAC Case Open tool at the following website:

http://www.cisco.com/tac/caseopen

Contacting TAC by Telephone

If you have a priority level 1 (P1) or priority level 2 (P2) problem, contact TAC by telephone and immediately open a case. To obtain a directory of toll-free numbers for your country, go to the following website:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

P1 and P2 level problems are defined as follows:

- P1—Your production network is down, causing a critical impact to business operations if service is not restored quickly. No workaround is available.
- P2—Your production network is severely degraded, affecting significant aspects of your business operations. No workaround is available.

Using Cisco IOS Software

This chapter provides helpful tips for understanding and configuring Cisco IOS software using the command-line interface (CLI). It contains the following sections:

- Understanding Command Modes
- Getting Help
- Using the no and default Forms of Commands
- Saving Configuration Changes
- Filtering Output from the show and more Commands
- Identifying Supported Platforms

For an overview of Cisco IOS software configuration, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information on the conventions used in the Cisco IOS software documentation set, see the chapter "About Cisco IOS Software Documentation" located at the beginning of this book.

Understanding Command Modes

You use the CLI to access Cisco IOS software. Because the CLI is divided into many different modes, the commands available to you at any given time depend on the mode you are currently in. Entering a question mark (**?**) at the CLI prompt allows you to obtain a list of commands available for each command mode.

When you log in to the CLI, you are in user EXEC mode. User EXEC mode contains only a limited subset of commands. To have access to all commands, you must enter privileged EXEC mode, normally by using a password. From privileged EXEC mode you can issue any EXEC command—user or privileged mode—or you can enter global configuration mode. Most EXEC commands are one-time commands. For example, **show** commands show important status information, and **clear** commands clear counters or interfaces. The EXEC commands are not saved when the software reboots.

Configuration modes allow you to make changes to the running configuration. If you later save the running configuration to the startup configuration, these changed commands are stored when the software is rebooted. To enter specific configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and a variety of other modes, such as protocol-specific modes.

ROM monitor mode is a separate mode used when the Cisco IOS software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode.

Table 1 describes how to access and exit various common command modes of the Cisco IOS software. It also shows examples of the prompts displayed for each mode.

Table 1 Accessing and Exiting Command Modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** EXEC command.	`Router#`	To return to user EXEC mode, use the **disable** command.
Global configuration	From privileged EXEC mode, use the **configure terminal** privileged EXEC command.	`Router(config)#`	To return to privileged EXEC mode from global configuration mode, use the **exit** or **end** command, or press **Ctrl-Z**.
Interface configuration	From global configuration mode, specify an interface using an **interface** command.	`Router(config-if)#`	To return to global configuration mode, use the **exit** command. To return to privileged EXEC mode, use the **end** command, or press **Ctrl-Z**.
ROM monitor	From privileged EXEC mode, use the **reload** EXEC command. Press the **Break** key during the first 60 seconds while the system is booting.	`>`	To exit ROM monitor mode, use the **continue** command.

For more information on command modes, refer to the "Using the Command-Line Interface" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Getting Help

Entering a question mark (**?**) at the CLI prompt displays a list of commands available for each command mode. You can also get a list of keywords and arguments associated with any command by using the context-sensitive help feature.

To get help specific to a command mode, a command, a keyword, or an argument, use one of the following commands:

Command	Purpose
help	Provides a brief description of the help system in any command mode.
*abbreviated-command-entry***?**	Provides a list of commands that begin with a particular character string. (No space between command and question mark.)
abbreviated-command-entry<**Tab**>	Completes a partial command name.
?	Lists all commands available for a particular command mode.
command **?**	Lists the keywords or arguments that you must enter next on the command line. (Space between command and question mark.)

Example: How to Find Command Options

This section provides an example of how to display syntax for a command. The syntax can consist of optional or required keywords and arguments. To display keywords and arguments for a command, enter a question mark (**?**) at the configuration prompt or after entering part of a command followed by a space. The Cisco IOS software displays a list and brief description of available keywords and arguments. For example, if you were in global configuration mode and wanted to see all the keywords or arguments for the **arap** command, you would type **arap ?**.

The <cr> symbol in command help output stands for "carriage return." On older keyboards, the carriage return key is the Return key. On most modern keyboards, the carriage return key is the Enter key. The <cr> symbol at the end of command help output indicates that you have the option to press **Enter** to complete the command and that the arguments and keywords in the list preceding the <cr> symbol are optional. The <cr> symbol by itself indicates that no more arguments or keywords are available and that you must press **Enter** to complete the command.

Table 2 shows examples of how you can use the question mark (**?**) to assist you in entering commands. The table steps you through configuring an IP address on a serial interface on a Cisco 7206 router that is running Cisco IOS Release 12.0(3).

Table 2 How to Find Command Options

Command	Comment

`Router> enable`
`Password: <password>`
`Router#`
	Enter the **enable** command and password to access privileged EXEC commands. You are in privileged EXEC mode when the prompt changes to `Router#`.

`Router# configure terminal`
`Enter configuration commands, one per line. End with CNTL/Z.`
`Router(config)#`
	Enter the **configure terminal** privileged EXEC command to enter global configuration mode. You are in global configuration mode when the prompt changes to `Router(config)#`.

`Router(config)# interface serial ?`
`  <0-6>  Serial interface number`
`Router(config)# interface serial 4 ?`
`  /`
`Router(config)# interface serial 4/ ?`
`  <0-3>  Serial interface number`
`Router(config)# interface serial 4/0`
`Router(config-if)#`
	Enter interface configuration mode by specifying the serial interface that you want to configure using the **interface serial** global configuration command.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter the serial interface slot number and port number, separated by a forward slash.
	You are in interface configuration mode when the prompt changes to `Router(config-if)#`.

Table 2 How to Find Command Options (continued)

Command	Comment
Router(config-if)# **?** Interface configuration commands: . . . ip Interface Internet Protocol config commands keepalive Enable keepalive lan-name LAN Name command llc2 LLC2 Interface Subcommands load-interval Specify interval for load calculation for an interface locaddr-priority Assign a priority group logging Configure logging for interface loopback Configure internal loopback on an interface mac-address Manually set interface MAC address mls mls router sub/interface commands mpoa MPOA interface configuration commands mtu Set the interface Maximum Transmission Unit (MTU) netbios Use a defined NETBIOS access list or enable name-caching no Negate a command or set its defaults nrzi-encoding Enable use of NRZI encoding ntp Configure NTP . . . Router(config-if)#	Enter **?** to display a list of all the interface configuration commands available for the serial interface. This example shows only some of the available interface configuration commands.
Router(config-if)# **ip ?** Interface IP configuration subcommands: access-group Specify access control for packets accounting Enable IP accounting on this interface address Set the IP address of an interface authentication authentication subcommands bandwidth-percent Set EIGRP bandwidth limit broadcast-address Set the broadcast address of an interface cgmp Enable/disable CGMP directed-broadcast Enable forwarding of directed broadcasts dvmrp DVMRP interface commands hello-interval Configures IP-EIGRP hello interval helper-address Specify a destination address for UDP broadcasts hold-time Configures IP-EIGRP hold time . . . Router(config-if)# **ip**	Enter the command that you want to configure for the interface. This example uses the **ip** command. Enter **?** to display what you must enter next on the command line. This example shows only some of the available interface IP configuration commands.

Table 2 How to Find Command Options (continued)

Command	Comment

`Router(config-if)# ip address ?`
`  A.B.C.D  IP address`
`  negotiated  IP Address negotiated over PPP`
`Router(config-if)# ip address`
	Enter the command that you want to configure for the interface. This example uses the **ip address** command.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP address or the **negotiated** keyword.
	A carriage return (<cr>) is not displayed; therefore, you must enter additional keywords or arguments to complete the command.

`Router(config-if)# ip address 172.16.0.1 ?`
`  A.B.C.D  IP subnet mask`
`Router(config-if)# ip address 172.16.0.1`
	Enter the keyword or argument you want to use. This example uses the 172.16.0.1 IP address.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP subnet mask.
	A <cr> is not displayed; therefore, you must enter additional keywords or arguments to complete the command.

`Router(config-if)# ip address 172.16.0.1 255.255.255.0 ?`
`  secondary  Make this IP address a secondary address`
`  <cr>`
`Router(config-if)# ip address 172.16.0.1 255.255.255.0`
	Enter the IP subnet mask. This example uses the 255.255.255.0 IP subnet mask.
	Enter **?** to display what you must enter next on the command line. In this example, you can enter the **secondary** keyword, or you can press **Enter**.
	A <cr> is displayed; you can press **Enter** to complete the command, or you can enter another keyword.

`Router(config-if)# ip address 172.16.0.1 255.255.255.0` `Router(config-if)#`	In this example, Enter is pressed to complete the command.

Using the no and default Forms of Commands

Almost every configuration command has a **no** form. In general, use the **no** form to disable a function. Use the command without the **no** keyword to reenable a disabled function or to enable a function that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** command; to reenable IP routing, use the **ip routing** command. The Cisco IOS software command reference publications provide the complete syntax for the configuration commands and describe what the **no** form of a command does.

Configuration commands also can have a **default** form, which returns the command settings to the default values. Most commands are disabled by default, so in such cases using the **default** form has the same result as using the **no** form of the command. However, some commands are enabled by default and

have variables set to certain default values. In these cases, the **default** form of the command enables the command and sets the variables to their default values. The Cisco IOS software command reference publications describe the effect of the **default** form of a command if the command functions differently than the **no** form.

Saving Configuration Changes

Use the **copy system:running-config nvram:startup-config** command to save your configuration changes to the startup configuration so that the changes will not be lost if the software reloads or a power outage occurs. For example:

```
Router# copy system:running-config nvram:startup-config
Building configuration...
```

It might take a minute or two to save the configuration. After the configuration has been saved, the following output appears:

```
[OK]
Router#
```

On most platforms, this task saves the configuration to NVRAM. On the Class A Flash file system platforms, this task saves the configuration to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Filtering Output from the show and more Commands

In Cisco IOS Release 12.0(1)T and later releases, you can search and filter the output of **show** and **more** commands. This functionality is useful if you need to sort through large amounts of output or if you want to exclude output that you need not see.

To use this functionality, enter a **show** or **more** command followed by the "pipe" character (|); one of the keywords **begin**, **include**, or **exclude**; and a regular expression on which you want to search or filter (the expression is case-sensitive):

command | {**begin** | **include** | **exclude**} *regular-expression*

The output matches certain lines of information in the configuration file. The following example illustrates how to use output modifiers with the **show interface** command when you want the output to include only lines in which the expression "protocol" appears:

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

For more information on the search and filter functionality, refer to the "Using the Command-Line Interface" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.2.

Identifying Supported Platforms

Cisco IOS software is packaged in feature sets consisting of software images that support specific platforms. The feature sets available for a specific platform depend on which Cisco IOS software images are included in a release. To identify the set of software images available in a specific release or to find out if a feature is available in a given Cisco IOS software image, see the following sections:

- Using Feature Navigator
- Using Software Release Notes

Using Feature Navigator

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a particular set of features and which features are supported in a particular Cisco IOS image.

Feature Navigator is available 24 hours a day, 7 days a week. To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, e-mail the Contact Database Administration group at cdbadmin@cisco.com. If you do not have an account on Cisco.com, go to http://www.cisco.com/register and follow the directions to establish an account.

To use Feature Navigator, you must have a JavaScript-enabled web browser such as Netscape 3.0 or later, or Internet Explorer 4.0 or later. Internet Explorer 4.0 always has JavaScript enabled. To enable JavaScript for Netscape 3.x or Netscape 4.x, follow the instructions provided with the web browser. For JavaScript support and enabling instructions for other browsers, check with the browser vendor.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. You can access Feature Navigator at the following URL:

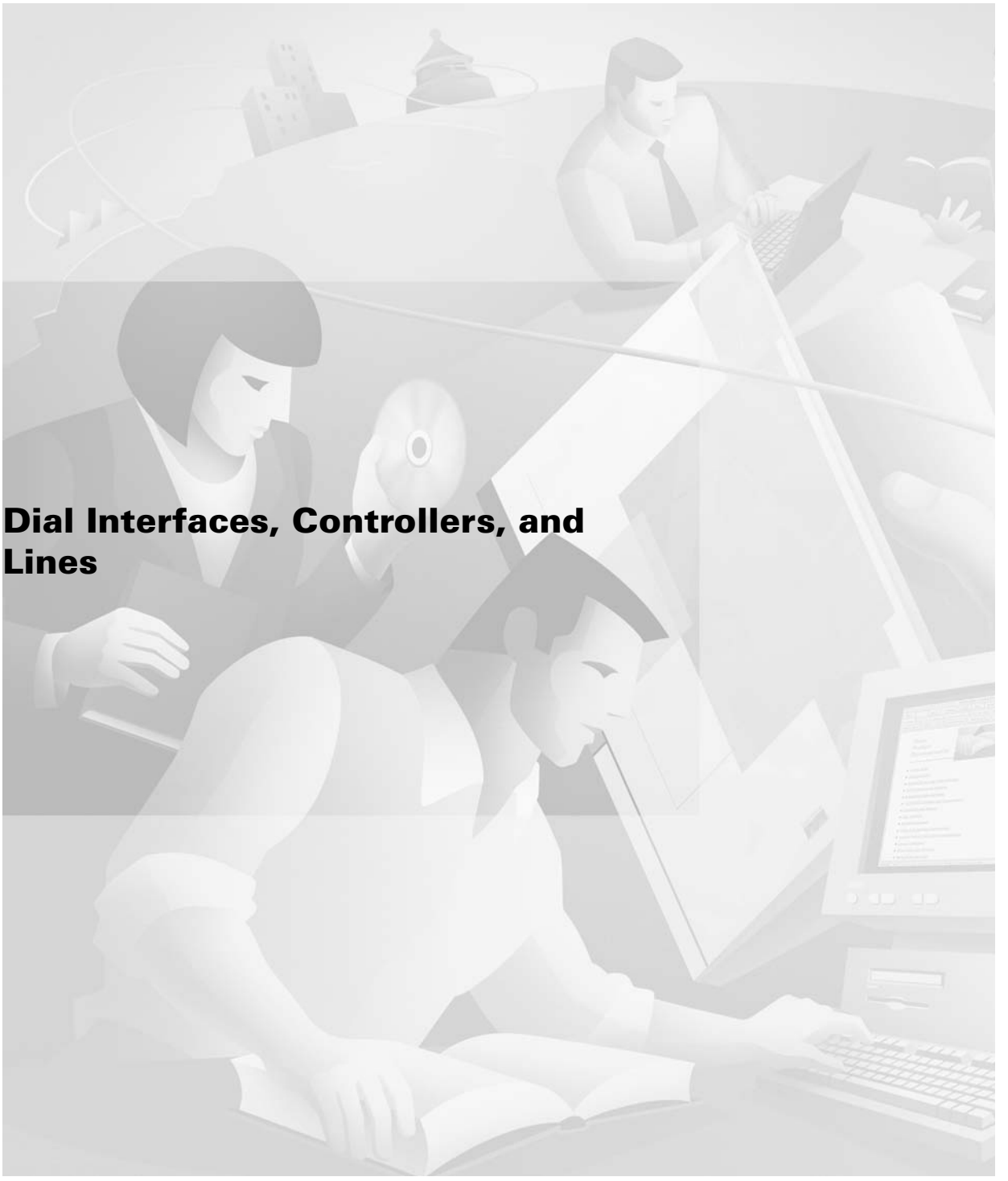
http://www.cisco.com/go/fn

Using Software Release Notes

Cisco IOS software releases include release notes that provide the following information:

- Platform support information
- Memory recommendations
- Microcode support information
- Feature set tables
- Feature descriptions
- Open and resolved severity 1 and 2 caveats for all platforms

Release notes are intended to be release-specific for the most current release, and the information provided in these documents may not be cumulative in providing information about features that first appeared in previous releases.



Dial Interfaces, Controllers, and Lines

Overview of Dial Interfaces, Controllers, and Lines

This chapter describes the different types of software constructs, interfaces, controllers, channels, and lines that are used for dial-up remote access. It includes the following main sections:

- Cisco IOS Dial Components
- Logical Constructs
- Logical Interfaces
- Circuit-Switched Digital Calls
- T1 and E1 Controllers
- Non-ISDN Channelized T1 and Channelized E1 Lines
- ISDN Service
- Line Types
- Encapsulation Types

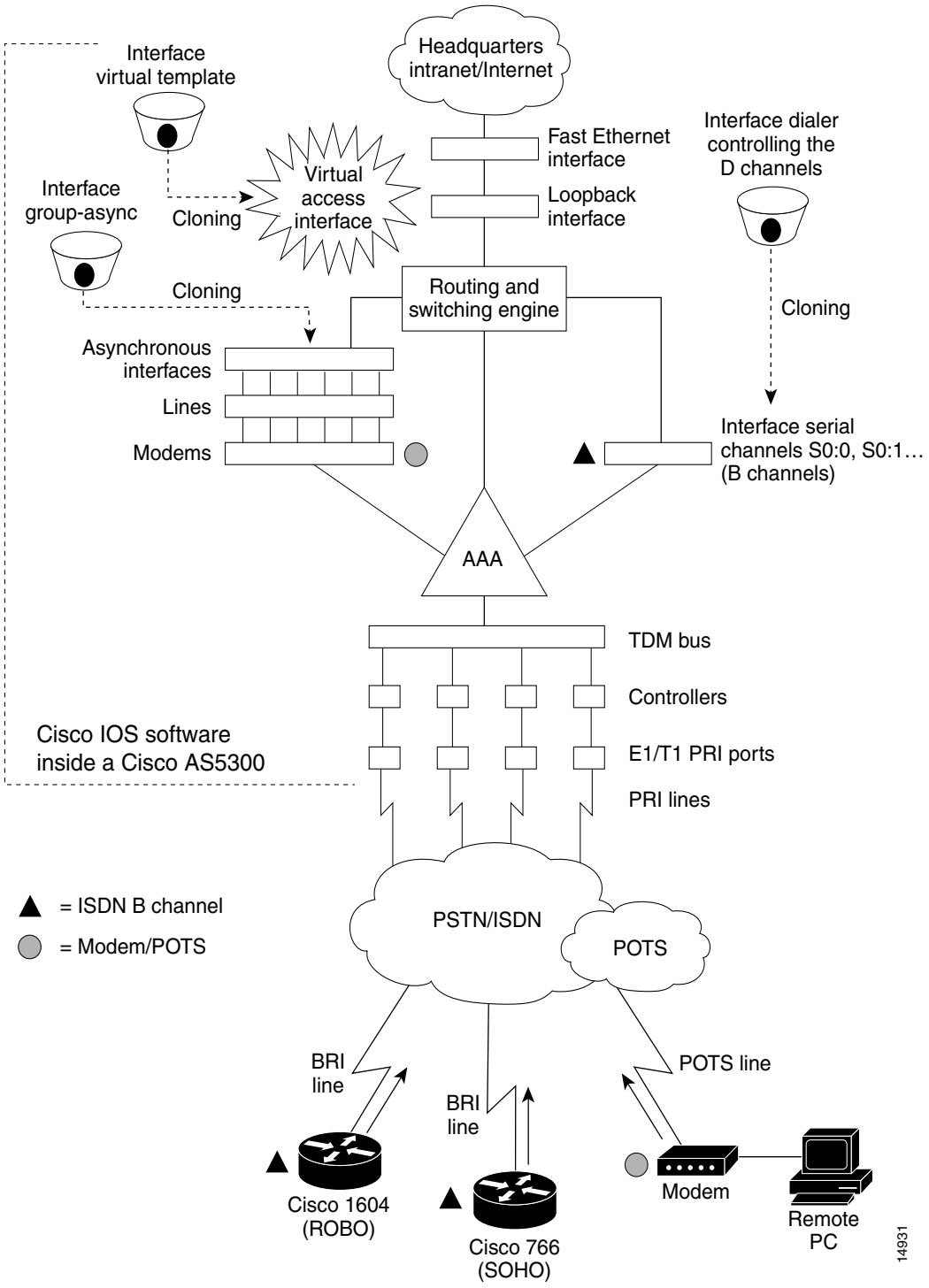
For a complete description of the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Cisco IOS Dial Components

Different components inside Cisco IOS software work together to enable remote clients to dial in and send packets. Figure 2 shows one Cisco AS5300 access server that is receiving calls from a remote office, branch office (ROBO); small office, home office (SOHO); and modem client.

Depending on your network scenario, you may encounter all of the components in Figure 2. For example, you might decide to create a virtual IP subnet by using a loopback interface. This step saves address space. Virtual subnets can exist inside devices that you advertise to your backbone. In turn, IP packets get relayed to remote PCs, which route back to the central site.

Figure 2 Cisco IOS Dial Universe



Logical Constructs

A logical construct stores core protocol characteristics to assign to physical interfaces. No data packets are forwarded to a logical construct. Cisco uses three types of logical constructs in its access servers and routers. These constructs are described in the following sections:

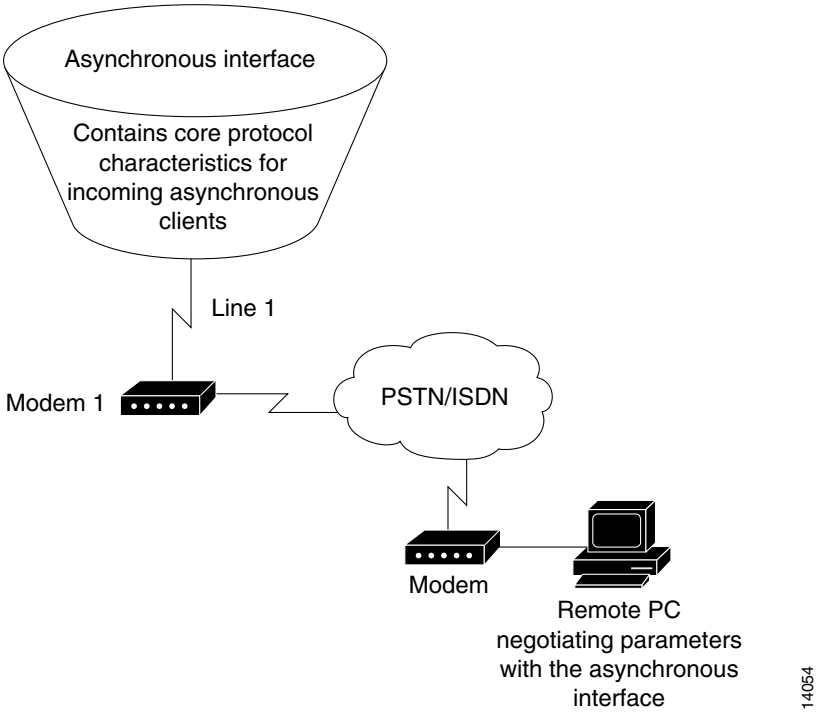
- Asynchronous Interfaces
- Group Asynchronous Interfaces
- Virtual Template Interfaces

Asynchronous Interfaces

An asynchronous interface assigns network protocol characteristics to remote asynchronous clients that are dialing in through physical terminal lines and modems. (See Figure 3.)

Use the **interface async** command to create and configure an asynchronous interface.

Figure 3 Logical Construct for an Asynchronous Interface



To enable clients to dial in, you must configure two asynchronous components: asynchronous lines and asynchronous interfaces. Asynchronous interfaces correspond to physical terminal lines. For example, asynchronous interface 1 corresponds to tty line 1.

Commands entered in asynchronous interface mode configure protocol-specific parameters for asynchronous interfaces, whereas commands entered in line configuration configure the physical aspects for the same port.

Specifically, you configure asynchronous interfaces to support PPP connections. An asynchronous interface on an access server or router can be configured to support the following functions:

- Network protocol support such as IP, Internet Protocol Exchange (IPX), or AppleTalk
- Encapsulation support (such as PPP)
- IP client addressing options (default or dynamic)
- IPX network addressing options
- PPP authentication
- ISDN BRI and PRI configuration

For additional information about configuring asynchronous interfaces, see the chapter "Configuring Asynchronous Lines and Interfaces."

Group Asynchronous Interfaces

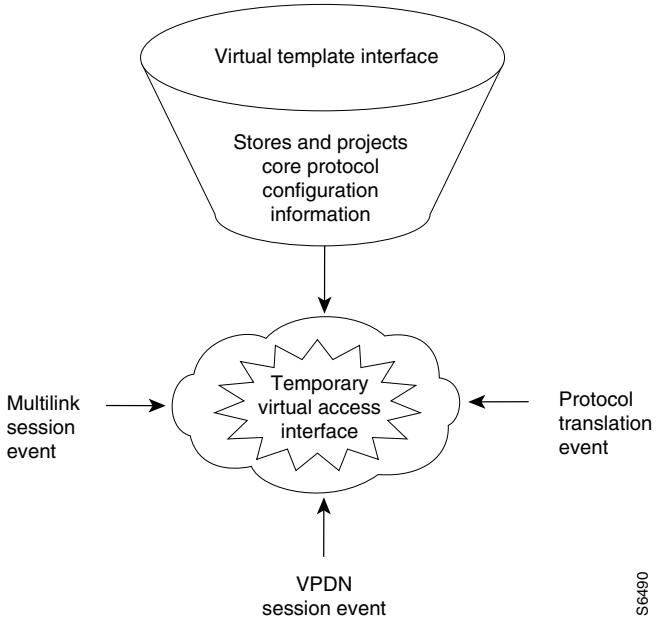
A group asynchronous interface is a parent interface that stores core protocol characteristics and projects them to a specified range of asynchronous interfaces. Asynchronous interfaces clone protocol information from group asynchronous interfaces. No data packets arrive in a group asynchronous interface. By setting up a group asynchronous interface, you also eliminate the need to repeatedly configure identical configuration information across several asynchronous interfaces.

See the "Overview of Modem Interfaces" chapter for more information about group asynchronous interfaces.

Virtual Template Interfaces

A virtual template interface stores protocol configuration information for virtual access interfaces and protocol translation sessions. (See Figure 4.)

Figure 4 Logical Construct for a Virtual Template Interface



Templates for Virtual Access Interfaces

Virtual templates project configuration information to temporary virtual access interfaces triggered by multilink or virtual private dial-up network (VPDN) session events. When a virtual access interface is triggered, the configuration attributes in the virtual template are cloned and the negotiated parameters are applied to the connection.

The following example shows a virtual template interface on a Cisco 7206 router, which is used as a home gateway in a VPDN scenario:

```
Router# configure terminal
Router(config)# interface virtual-template 1
Router(config-if)# ip unnumbered ethernet 2/1
Router(config-if)# peer default ip address pool cisco-pool
Router(config-if)# ppp authentication chap pap
Router(config-if)# exit
Router(config)# vpdn enable
Router(config)# vpdn incoming isp cisco.com virtual-template 1
```

Templates for Protocol Translation

Virtual templates are used to simplify the process of configuring protocol translation to tunnel PPP or Serial Line Internet Protocol (SLIP) across X.25, TCP, and LAT networks. You can create a virtual interface template using the **interface virtual-template** command, and you can use it for one-step and two-step protocol translation. When a user dials in through a vty line and a tunnel connection is established, the router clones the attributes of the virtual interface template onto a *virtual access interface*. This virtual access interface is a temporary interface that supports the protocol configuration specified in the virtual interface template. This virtual access interface is created dynamically and lasts only as long as the tunnel session is active.

The virtual template in the following example explicitly specifies PPP encapsulation. The translation is from X.25 to PPP, which enables tunneling of PPP across an X.25 network.

```
Router# configure terminal
Router(config)# interface virtual-template 1
Router(config-if)# ip unnumbered ethernet 0
Router(config-if)# peer default ip address 172.18.2.131
Router(config-if)# encapsulation ppp
Router(config-if)# exit
Router(config)# translate x25 5555678 virtual-template 1
```

For more information, refer to the chapter “Configuring Protocol Translation and Virtual Asynchronous Devices” in the *Cisco IOS Terminal Services Configuration Guide*.

Logical Interfaces

A logical interface receives and sends data packets and controls physical interfaces. Cisco IOS software provides three logical interfaces used for dial access. These interfaces are described in the following sections:

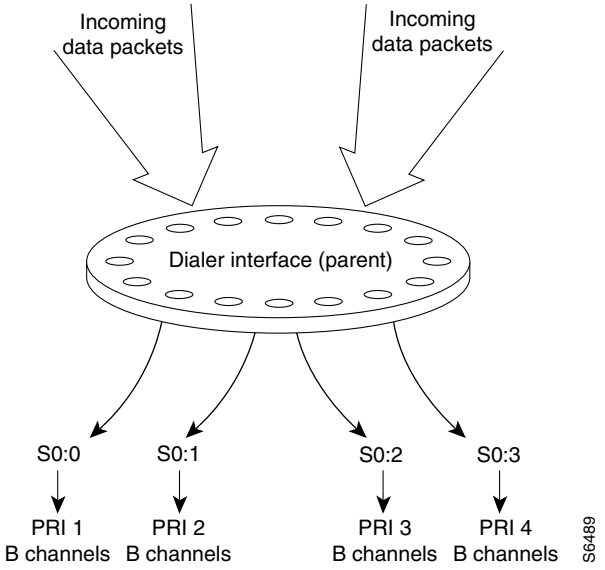
- Dialer Interfaces
- Virtual Access Interfaces
- Virtual Asynchronous Interfaces

Dialer Interfaces

A dialer interface is a parent interface that stores and projects protocol configuration information that is common to all data (D) channels that are members of a dialer rotary group. Data packets pass through dialer interfaces, which in turn initiate dialing for inbound calls. In most cases, D channels get their core protocol intelligence from dialer interfaces.

Figure 5 shows packets coming into a dialer interface, which contains the configuration parameters common to four D channels (shown as S0:0, S0:1, S0:2, and S0:3). All the D channels are members of the same rotary group. Without the dialer interface configuration, each D channel must be manually configured with identical properties. Dialer interfaces condense and streamline the configuration process.

Figure 5 Dialer Interface and Its Neighboring Components



A dialer interface is user configurable and linked to individual B channels, where it delivers data packets to their physical destinations. Dialer interfaces seize physical interfaces to cause packet delivery. If a dialer interface engages in a multilink session, a dialer interface is in control of a virtual access interface, which in turn controls S0:3 or chassis 2 S0:3, for example. A dialer interface is created with the **interface dialer** global configuration command.

The following example shows a fully configured dialer interface:

```
Router# configure terminal
Router(config)# interface dialer 0
Router(config-if)# ip unnumbered loopback 0
Router(config-if)# no ip mroute-cache
Router(config-if)# encapsulation ppp
Router(config-if)# peer default ip address pool dialin_pool
Router(config-if)# dialer in-band
Router(config-if)# dialer-group 1
Router(config-if)# no fair-queue
Router(config-if)# no cdp enable
Router(config-if)# ppp authentication chap pap callin
Router(config-if)# ppp multilink
```

All the D channels are members of rotary group 1.

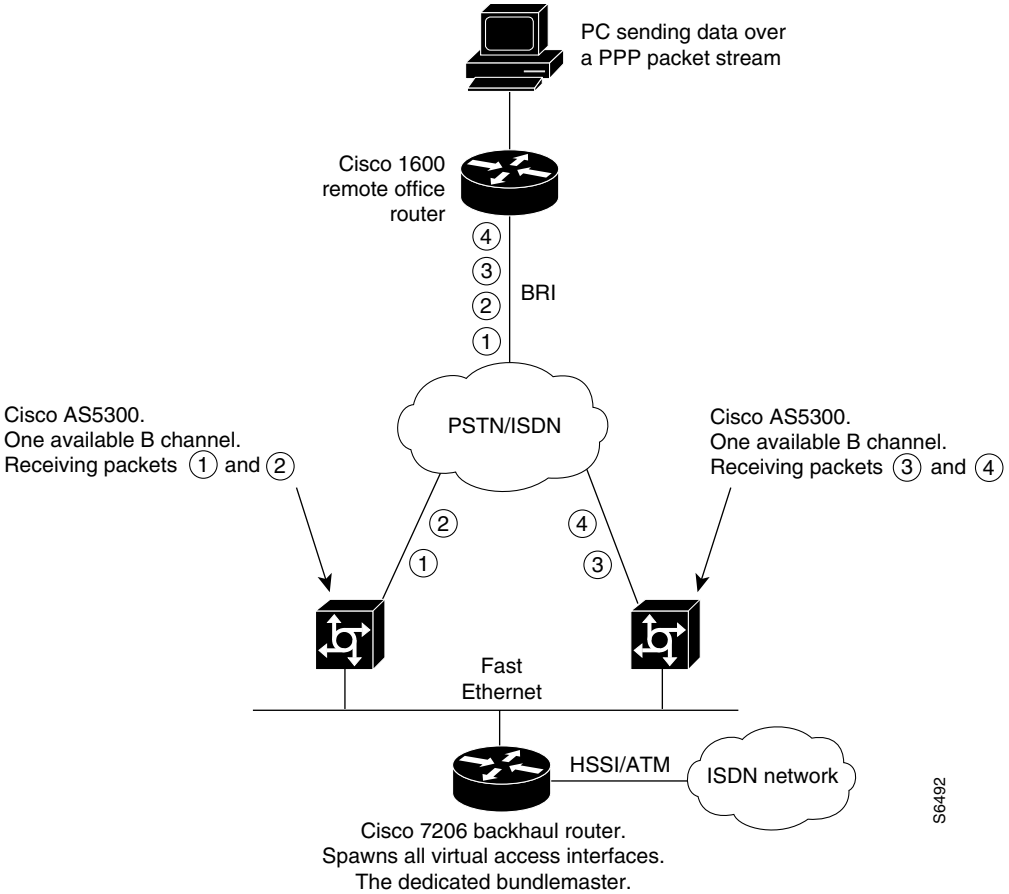
Virtual Access Interfaces

A virtual access interface is a temporary interface that is spawned to terminate incoming PPP streams that have no physical connections. PPP streams, Layer 2 Forwarding Protocol (L2F), and Layer 2 Tunnel Protocol (L2TP) frames that come in on multiple B channels are reassembled on virtual access interfaces. These access interfaces are constructs used to terminate packets.

Virtual access interfaces obtain their set of instructions from virtual interface templates. The attributes configured in virtual templates are projected or cloned to a virtual access interfaces. Virtual access interfaces are not directly user configurable. These interfaces are created dynamically and last only as long as the tunnels or multilink sessions are active. After the sessions end, the virtual access interfaces disappear.

Figure 6 shows how a virtual access interface functions to accommodate a multilink session event. Two physical interfaces on two different access servers are participating in one multilink call from a remote PC. However, each Cisco AS5300 access server has only one B channel available to receive a call. All other channels are busy. Therefore all four packets are equally dispersed across two separate B channels and two access servers. Each Cisco AS5300 access server receives only half the total packets. A virtual access interface is dynamically spawned upstream on a Cisco 7206 backhaul router to receive the multilink protocol, track the multilink frames, and reassemble the packets. The Cisco 7206 router is configured to be the bundle master, which performs all packet assembly and reassembly for both Cisco AS5300 access servers.

Figure 6 Virtual Access Interfaces Used for Multichassis Multilink Session Events

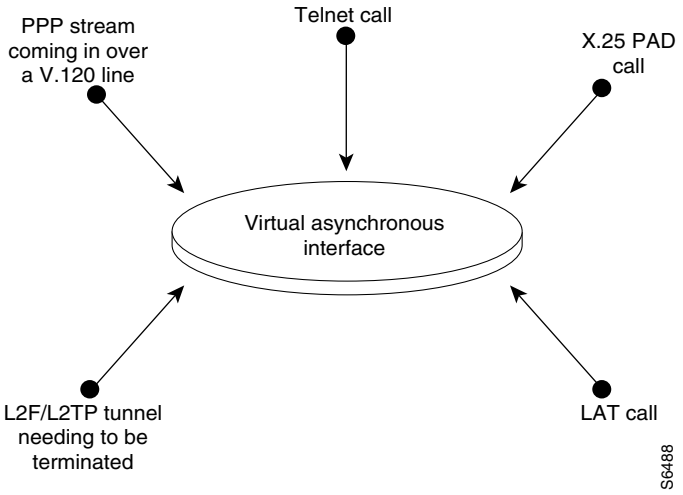


Virtual Asynchronous Interfaces

A virtual asynchronous interface is created on demand to support calls that enter the router through a nonphysical interface. For example, asynchronous character stream calls terminate or land on nonphysical interfaces. These types of calls include inbound Telnet, LAT, PPP over character-oriented protocols (such as V.120 or X.25), and LAPB-TA and PAD calls. A virtual asynchronous interface is also used to terminate L2F/L2TP tunnels, which are often traveling companions with Multilink protocol sessions. Virtual asynchronous interfaces are not user configurable; rather, they are dynamically created and torn down on demand. A virtual asynchronous line is used to access a virtual asynchronous interface.

Figure 7 shows a variety of calls that are terminating on a virtual asynchronous interface. After the calls end, the interface is torn down.

Figure 7 Asynchronous Character Stream Calls Terminating on a Virtual Asynchronous Interface

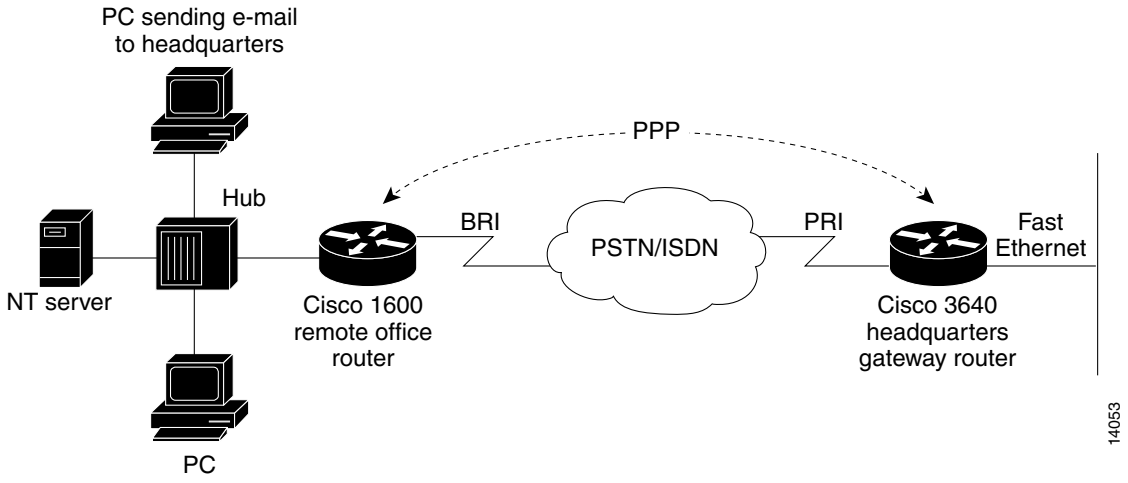


Circuit-Switched Digital Calls

Circuit-switched digital calls are usually ISDN 56-kbps or 64-kbps data calls that use PPP. These calls are initiated by an ISDN router, access server, or terminal adapter that is connected to a client workstation. Individual synchronous serial digital signal level 0 (DS0) bearer (B) channels are used to transport circuit-switched digital calls across WANs. These calls do not transmit across “old world” lines.

Figure 8 shows a Cisco 1600 series remote office router dialing in to a Cisco 3640 router positioned at a headquarters gateway.

Figure 8 Remote Office LAN Dialing In to Headquarters



T1 and E1 Controllers

Cisco controllers negotiate the following parameters between an access server and a central office: line coding, framing, clocking, DS0/time-slot provisioning, and signaling.

Time slots are provisioned to meet the needs of particular network scenarios. T1 controllers have 24 time slots, and E1 controllers have 30 time slots. To support traffic flow for one ISDN PRI line in a T1 configuration, use the **pri-group** command. To support traffic flow for analog calls over a channelized E1 line with recEive and transMit (E&M—also ear and mouth) signaling, use the **cas-group 1 timeslots 1-30 type e&m-fgb** command. Most telephone companies do not support provisioning one trunk for different combinations of time-slot services, though this provisioning is supported on Cisco controllers. On a T1 controller, for example, time slots 1 to 10 could run PRI, time slots 11 to 20 could run channel-associated signaling (CAS), and time slots 21 to 24 could support leased-line grouping.

The following example configures one of four T1 controllers on a Cisco AS5300 access server:

```
Router# configure terminal
Router(config)# controller t1 ?
  <0-3>  Controller unit number
Router(config)# controller t1 0
Router(config-controller)# framing esf
Router(config-controller)# linecode b8zs
Router(config-controller)# clock source line primary
Router(config-controller)# pri-group timeslots 1-24
Router(config-controller)#
```

This example supports modem calls and circuit-switched digital calls over ISDN PRI.

Non-ISDN Channelized T1 and Channelized E1 Lines

A channelized T1 or channelized E1 line is an analog line that was originally intended to support analog voice calls, but has evolved to support analog data calls. ISDN is not sent across channelized T1 or E1 lines. Channelized T1 and channelized E1 lines are often referred to as CT1 and CE1. These channelized lines are found in “old world,” non-ISDN telephone networks.

The difference between traditional channelized lines (analog) and nonchannelized lines (ISDN) is that channelized lines have no built-in D channel. That is, all 24 channels on a T1 line carry only data. The signaling is in-band or associated to the data channels. Traditional channelized lines do not support digitized data calls (for example, BRI with 2B + D). Channelized lines support a variety of in-band signal types, such as ground start, loop start, wink start, immediate start, E&M, and R2.

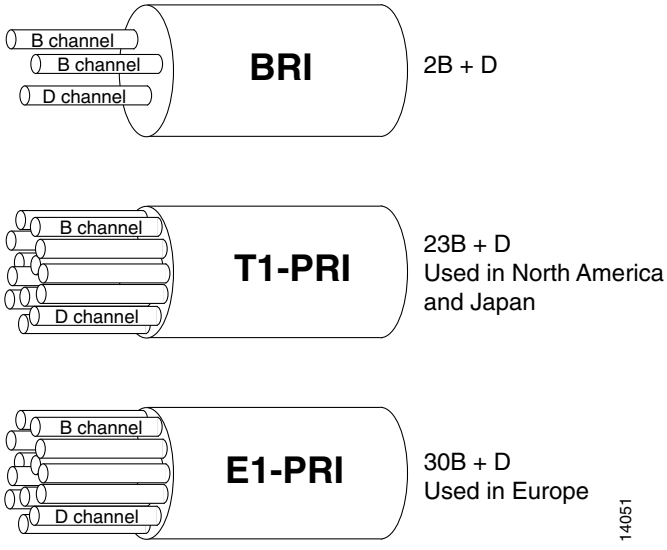
Signaling for channelized lines is configured with the **cas-group** controller configuration command. The following example configures E&M group B signaling on a T1 controller:

```
Router# configure terminal
Router(config)# controller t1 0
Router(config-controller)# cas-group 1 timeslots 1-24 type ?
  e&m-fgb              E & M Type II FGB
  e&m-fgd              E & M Type II FGD
  e&m-immediate-start  E & M Immediate Start
  fxs-ground-start     FXS Ground Start
  fxs-loop-start       FXS Loop Start
  r1-modified          R1 Modified
  sas-ground-start     SAS Ground Start
  sas-loop-start       SAS Loop Start
Router(config-controller)# cas-group 1 timeslots 1-24 type e&m-fgb
Router(config-controller)# framing esf
Router(config-controller)# clock source line primary
```

ISDN Service

Cisco routing devices support ISDN BRI and ISDN PRI. Both media types use B channels and D channels. Figure 9 shows how many B channels and D channels are assigned to each media type.

Figure 9 Logical Relationship of B Channels and D Channels



ISDN BRI

ISDN BRI operates over most of the copper twisted-pair telephone wiring in place. ISDN BRI delivers a total bandwidth of a 144 kbps via three separate channels. Two of the B channels operate at 64 kbps and are used to carry voice, video, or data traffic. The third channel, the D channel, is a 16-kbps signaling channel used to tell the Public Switched Telephone Network (PSTN) how to handle each of the B channels. ISDN BRI is often referred to as "2 B + D."

Enter the **interface bri** command to bring up and configure a single BRI interface, which is the overseer of the 2 B + D channels. The D channel is not user configurable.

The following example configures an ISDN BRI interface on a Cisco 1600 series router. The **isdn spid** command defines the service profile identifier (SPID) number for both B channels. The SPID number is assigned by the ISDN service provider. Not all ISDN lines have SPIDs.

```
Router# configure terminal

Router(config)# interface bri 0
Router(config-if)# isdn spid1 55598760101
Router(config-if)# isdn spid2 55598770101
Router(config-if)# isdn switch-type basic-ni
Router(config-if)# ip unnumbered ethernet 0
Router(config-if)# dialer map ip 172.168.37.40 name hq 5552053
Router(config-if)# dialer load-threshold 70
Router(config-if)# dialer-group 1
Router(config-if)# encapsulation ppp
Router(config-if)# ppp authentication chap pap callin
Router(config-if)# ppp multilink
Router(config-if)# no shutdown
```

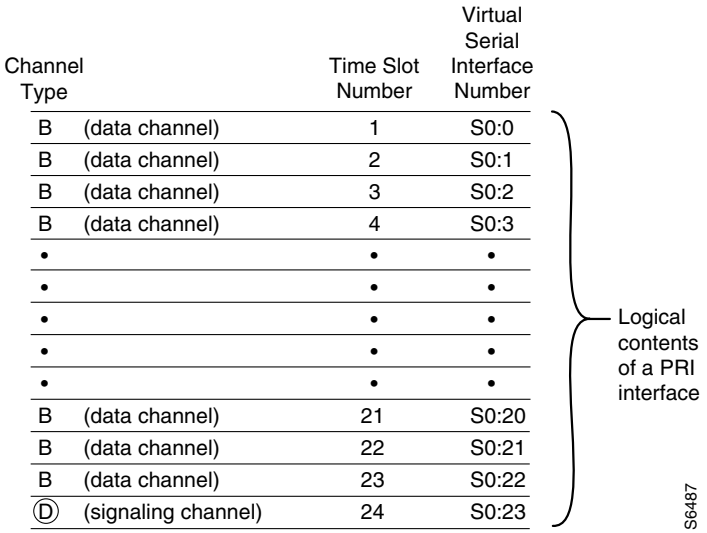
ISDN PRI

ISDN PRI is designed to carry large numbers of incoming ISDN calls at point of presences (POPs) and other large central site locations. All the reliability and performance of ISDN BRI applies to ISDN PRI, but ISDN PRI has 23 B channels running at 64 kbps each and a shared 64 kbps D channel that carries signaling traffic. ISDN PRI is often referred to as "23 B + D" (North America and Japan) or "30 B + D" (rest of the world).

The D channel notifies the central office switch to send the incoming call to particular timeslots on the Cisco access server or router. Each one of the B channels carries data or voice. The D channel carries signaling for the B channels. The D channel identifies if the call is a circuit-switched digital call or an analog modem call. Analog modem calls are decoded and then sent to the onboard modems. Circuit-switched digital calls are directly relayed to the ISDN processor in the router. Enter the **interface serial** command to bring up and configure the D channel, which is user configurable.

Figure 10 shows the logical contents of an ISDN PRI interface used in a T1 network configuration. The logical contents include 23 B channels, 1 D channel, 24 time slots, and 24 virtual serial interfaces (total number of B + D channels).

Figure 10 Logical Relationship of ISDN PRI Components for T1



The following example is for a Cisco AS5300 access server. It configures one T1 controller for ISDN PRI, then configures the neighboring D channel (interface serial 0:23). Controller T1 0 and interface serial 0:23 are both assigned to the first PRI port. The second PRI port is assigned to controller T1 1 and interface serial 1:23, and so on. The second PRI port configuration is not shown in this example. This Cisco AS5300 access server is used as part of a stack group dial-in solution for an Internet service provider.

```
Router# configure terminal

Router(config)# controller t1 0
Router(config-controller)# framing esf
Router(config-controller)# linecode b8zs
Router(config-controller)# clock source line primary
Router(config-controller)# pri-group timeslots 1-24
Router(config-controller)# exit
Router(config)# interface serial 0:23
Router(config-if)# ip unnumbered Loopback 0
Router(config-if)# ip accounting output-packets
Router(config-if)# no ip mroute-cache
Router(config-if)# encapsulation ppp
Router(config-if)# isdn incoming-voice modem
Router(config-if)# dialer-group 1
Router(config-if)# no fair-queue
Router(config-if)# compress stac
Router(config-if)# no cdp enable
Router(config-if)# ppp authentication chap
Router(config-if)# ppp multilink
Router(config-if)# netbios nbf
```

Line Types

This section describes the different line types used for dial access. It also describes the relationship between lines and interfaces.

Note Cisco devices have four types of lines: console, auxiliary, asynchronous, and virtual terminal. Different routers have different numbers of these line types. Refer to the hardware and software configuration guides that shipped with your device for exact configurations.

Table 3 shows the types of lines that can be configured.

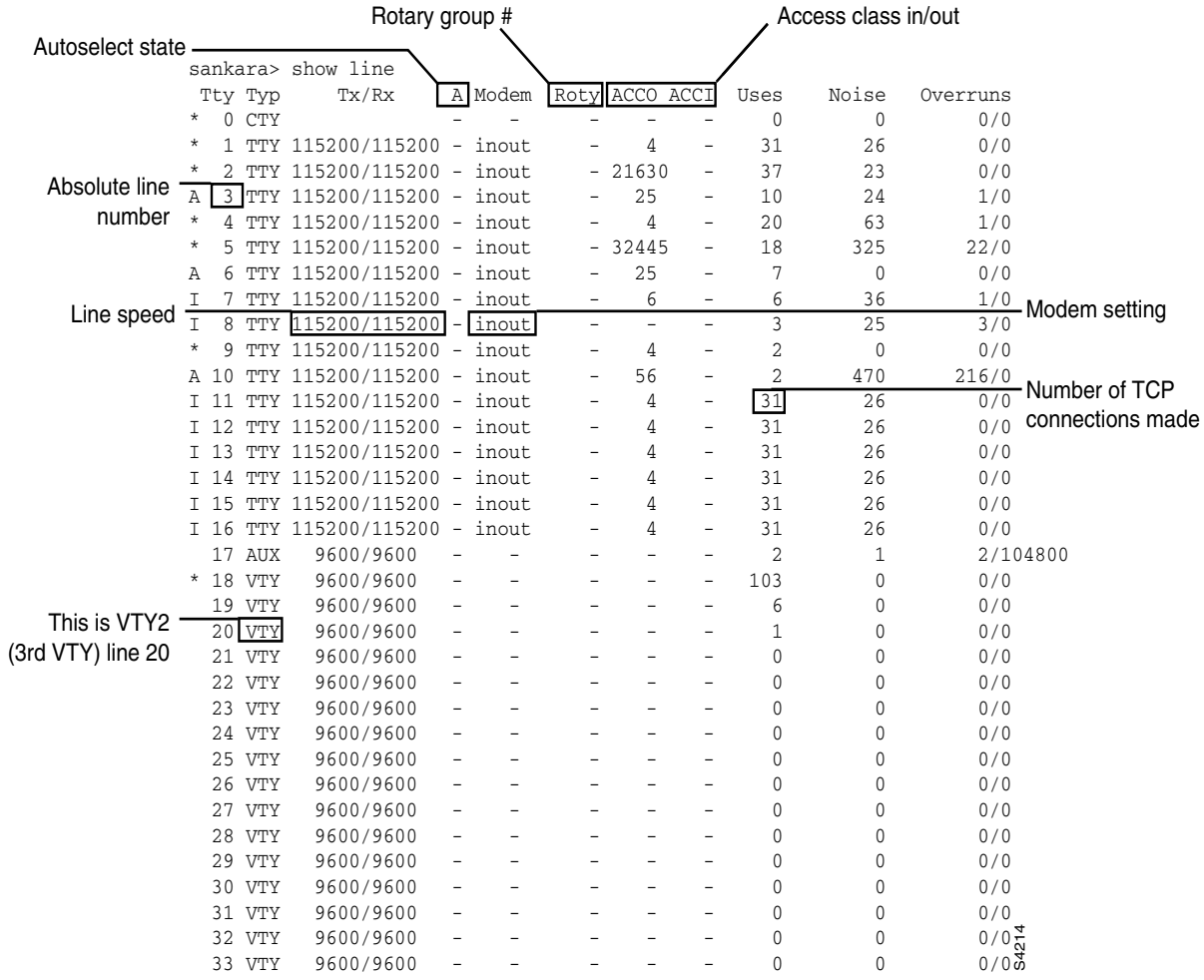
Table 3 Available Line Types

Line Type	Interface	Description	Numbering Rules
CON or CTY	Console	Typically used to log in to the router for configuration purposes.	Line 0.
AUX	Auxiliary	EIA/TIA-232 data terminal equipment (DTE) port used as a backup (tty) asynchronous port. Cannot be used as a second console port.	Last tty line number plus 1.
tty	Asynchronous	Same as asynchronous interface. Used typically for remote-node dial-in sessions that use such protocols as SLIP, PPP, AppleTalk Remote Access (ARA), and XRemote.	The numbering widely varies between platforms. This number is equivalent to the maximum number of modems or asynchronous interfaces supported by your access server or router.[1]
vty	Virtual asynchronous	Used for incoming Telnet, LAT, X.25 PAD, and protocol translation connections into synchronous ports (such as Ethernet and serial interfaces) on the router.	Last tty line number plus 2 through the maximum number of vty lines specified.[2]

1. Enter the **interface line tty ?** command to view the maximum number of tty lines supported.
2. Increase the number of vty lines on a router using the **line vty** global configuration command. Delete vty lines with the **no line vty** *line-number* command. The **line vty** command accepts any line number larger than 5 up to the maximum number of lines supported by your router with its current configuration. Enter the **interface line vty ?** command to view the maximum number of vty lines supported.

Use the **show line** command to see the status of each of the lines available on a router. (See Figure 11.)

Figure 11 Sample Show Line Output Showing CTY, tty, AUX, and vty Line Statistics




```
sankara> show line
 Tty Typ     Tx/Rx      A Modem  Roty ACCO ACCI   Uses    Noise   Overruns
*  0 CTY                 -    -     -    -    -      0        0        0/0
*  1 TTY 115200/115200 - inout     -    4    -     31       26        0/0
*  2 TTY 115200/115200 - inout     - 21630   -     37       23        0/0
A  3 TTY 115200/115200 - inout     -   25    -     10       24        1/0
*  4 TTY 115200/115200 - inout     -    4    -     20       63        1/0
*  5 TTY 115200/115200 - inout     - 32445   -     18      325       22/0
A  6 TTY 115200/115200 - inout     -   25    -      7        0        0/0
I  7 TTY 115200/115200 - inout     -    6    -      6       36        1/0
I  8 TTY 115200/115200 - inout     -    -    -      3       25        3/0
*  9 TTY 115200/115200 - inout     -    4    -      2        0        0/0
A 10 TTY 115200/115200 - inout     -   56    -      2      470      216/0
I 11 TTY 115200/115200 - inout     -    4    -     31       26        0/0
I 12 TTY 115200/115200 - inout     -    4    -     31       26        0/0
I 13 TTY 115200/115200 - inout     -    4    -     31       26        0/0
I 14 TTY 115200/115200 - inout     -    4    -     31       26        0/0
I 15 TTY 115200/115200 - inout     -    4    -     31       26        0/0
I 16 TTY 115200/115200 - inout     -    4    -     31       26        0/0
  17 AUX   9600/9600   -    -     -    -    -      2        1        2/104800
* 18 VTY   9600/9600   -    -     -    -    -    103        0        0/0
  19 VTY   9600/9600   -    -     -    -    -      6        0        0/0
  20 VTY   9600/9600   -    -     -    -    -      1        0        0/0
  21 VTY   9600/9600   -    -     -    -    -      0        0        0/0
  22 VTY   9600/9600   -    -     -    -    -      0        0        0/0
  23 VTY   9600/9600   -    -     -    -    -      0        0        0/0
  24 VTY   9600/9600   -    -     -    -    -      0        0        0/0
  25 VTY   9600/9600   -    -     -    -    -      0        0        0/0
  26 VTY   9600/9600   -    -     -    -    -      0        0        0/0
  27 VTY   9600/9600   -    -     -    -    -      0        0        0/0
  28 VTY   9600/9600   -    -     -    -    -      0        0        0/0
  29 VTY   9600/9600   -    -     -    -    -      0        0        0/0
  30 VTY   9600/9600   -    -     -    -    -      0        0        0/0
  31 VTY   9600/9600   -    -     -    -    -      0        0        0/0
  32 VTY   9600/9600   -    -     -    -    -      0        0        0/0
  33 VTY   9600/9600   -    -     -    -    -      0        0        0/0
```

Relationship Between Lines and Interfaces

The following sections describe the relationship between lines and interfaces:

- Asynchronous Interfaces and Physical Terminal Lines
- Synchronous Interfaces and Virtual Terminal Lines

Asynchronous Interfaces and Physical Terminal Lines

Asynchronous interfaces correspond to physical terminal lines. Commands entered in asynchronous interface mode let you configure protocol-specific parameters for asynchronous interfaces; commands entered in line configuration mode let you configure the physical aspects of the line port.

For example, to enable IP resources to dial in to a network through a Cisco 2500 series access server, configure the lines and asynchronous interfaces as follows.

- Configure the physical aspect of a line that leads to a port. You might enter the following commands to configure lines 1 through 16 (asynchronous physical terminal lines on a Cisco 2511 access server):

```
line 1 16
 login local
 modem inout
 speed 115200
 flowcontrol hardware
 ! Configures the line to autosense PPP; physical line attribute.
 autoselect ppp
```

- On asynchronous interface 1, you configure your protocol-specific commands. You might enter the following commands:

```
interface async 1
 encapsulation ppp
 async mode interactive
 async dynamic address
 async dynamic routing
 async default ip address 192.168.16.132
 ppp authentication chap
```

The remote node services SLIP, PPP, and XRemote are configured in asynchronous interface mode. ARA is configured in line configuration mode on virtual terminal lines or physical terminal lines.

Synchronous Interfaces and Virtual Terminal Lines

Virtual terminal lines provide access to the router through a synchronous interface. Virtual terminal lines do not correspond to synchronous interfaces in the same way that physical terminal lines correspond to asynchronous interfaces because vty lines are created dynamically on the router, whereas physical terminal lines are static physical ports. When a user connects to the router on a vty line, that user is connecting into a *virtual* port on an interface. You can have multiple virtual ports for each synchronous interface.

For example, several Telnet connections can be made to an interface (such as an Ethernet or serial interface).

The number of virtual terminal lines available on a router is defined using the **line vty** *number-of-lines* global configuration command.

Encapsulation Types

Synchronous serial interfaces default to High-Level Data Link Control (HDLC) encapsulation, and asynchronous serial interfaces default to SLIP encapsulation. Cisco IOS software provides a long list of encapsulation methods that can be set on the interface to change the default encapsulation method. See the *Cisco IOS Interface Command Reference* for a complete list and description of these encapsulation methods.

The following list summarizes the encapsulation commands available for serial interfaces used in dial configurations:

- **encapsulation frame-relay**—Frame Relay
- **encapsulation hdlc**—HDLC protocol
- **encapsulation lapb**—X.25 LAPB DTE operation
- **encapsulation ppp**—PPP
- **encapsulation slip**—SLIP

To use SLIP or PPP encapsulation, the router or access server must be configured with an IP routing protocol or with the **ip host-routing** command.

Configuring Asynchronous Lines and Interfaces

This chapter describes how to configure asynchronous line features in the following main sections:

- How to Configure Asynchronous Interfaces and Lines
- How to Configure Other Asynchronous Line and Interface Features
- Configuration Examples for Asynchronous Interfaces and Lines

Perform these tasks, as required, for your particular network.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

For a complete description of the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

How to Configure Asynchronous Interfaces and Lines

To configure an asynchronous interface, perform the tasks described in the following sections as required:

- Configuring a Typical Asynchronous Interface (As required)
- Creating a Group Asynchronous Interface (As required)
- Configuring Asynchronous Rotary Line Queueing (As required)
- Configuring Autoselect (As required)

Configuring a Typical Asynchronous Interface

To configure an asynchronous interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface async number`	Brings up a single asynchronous interface and enters interface configuration mode.
Step 2	`Router(config-if)# description description`	Provides a description for the interface.
Step 3	`Router(config-if)# ip address address mask`	Specifies an IP address.
Step 4	`Router(config-if)# encapsulation ppp`	Enables PPP to run on the asynchronous interfaces in the group.
Step 5	`Router(config-if)# async default routing`	Enables the router to pass routing updates to other routers over the AUX port configured as an asynchronous interface.
Step 6	`Router(config-if)# async mode dedicated`	Places a line into dedicated asynchronous mode using Serial Line Internet Protocol (SLIP) or PPP encapsulation.
Step 7	`Router(config-if)# dialer in-band`	Specifies that dial-on-demand routing (DDR) is to be supported.
Step 8	`Router(config-if)# dialer map protocol next-hop-address`	Configures a serial interface to call one or multiple sites or to receive calls from multiple sites.
Step 9	`Router(config-if)# dialer-group`	Controls access by configuring an interface to belong to a specific dialing group.
Step 10	`Router(config-if)# ppp authentication chap pap list-name`	Enables Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP) authentication on the interface. Replace the *list-name* variable with a specified authentication list name.[1]
Step 11	`Router(config-if)# exit`	Return to global configuration mode.

1. To create a string used to name the following list of authentication methods tried when a user logs in, refer to the **aaa authentication ppp** command. Authentication methods include RADIUS, TACACS+, and Kerberos.

The "Interface and Line Configuration Examples" and "Asynchronous Interface As the Only Network Interface Example" sections later in this chapter contain examples of how to configure an asynchronous interface.

Monitoring and Maintaining Asynchronous Connections

This section describes the following monitoring and maintenance tasks that you can perform on asynchronous interfaces:

- Monitoring and maintaining asynchronous activity
- Debugging asynchronous interfaces
- Debugging PPP

To monitor and maintain asynchronous activity, use the following commands in privileged EXEC mode as needed:

Command	Purpose
`Router# clear line line-number`	Returns a line to its idle state.
`Router# show async bootp`	Displays parameters that have been set for extended BOOTP requests.
`Router# show async status`	Displays statistics for asynchronous interface activity.
`Router# show line [line-number]`	Displays the status of asynchronous line connections.

To debug asynchronous interfaces, use the following debug command in privileged EXEC mode:

Command	Purpose
`Router# debug async {framing | state | packets}`	Displays errors, changes in interface state, and log input and output.

To debug PPP links, use the following debug commands in privileged EXEC mode as needed:

Command	Purpose
`Router# debug ppp negotiation`	Enables debugging of PPP protocol negotiation process.
`Router# debug ppp error`	Displays PPP protocol errors.
`Router# debug ppp packet`	Displays PPP packets sent and received.
`Router# debug ppp chap`	Displays errors encountered during remote or local system authentication.

Creating a Group Asynchronous Interface

Create a group asynchronous interface to project a set of core protocol characteristics to a range of asynchronous interfaces. Configuring the asynchronous interfaces as a group saves you time. Analog modem calls cannot enter the access server without this configuration.

To configure a group asynchronous interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface async number`	Brings up a single asynchronous interface and enters interface configuration mode.
Step 2	`Router(config-if)# ip unnumbered loopback number`	Configures the asynchronous interfaces as unnumbered and assigns the IP address of the loopback interface to them to conserve IP addresses.[1]
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP to run on the asynchronous interfaces in the group.

	Command	Purpose
Step 4	`Router(config-if)# async mode interactive`	Configures interactive mode on the asynchronous interface.
Step 5	`Router(config-if)# ppp authentication chap pap` *list-name*	Enables CHAP and PAP authentication on the interface. Replace the *list-name* variable with a specified authentication list name.[2]
Step 6	`Router(config-if)# peer default ip address pool` *poolname*	Assigns dial-in clients IP addresses from an address pool.[3]
Step 7	`Router(config-if)# no cdp enable`	Disables the Cisco Discovery Protocol (CDP) on the interface.
Step 8	`Router(config-if)# group-range` *low-end-of-range high-end-of-range*	Specifies the range of asynchronous interfaces to include in the group, which is usually equal to the number of modems you have in the access server.
Step 9	`Router(config-if)# exit`	Returns to global configuration mode.

1. You can also specify the Ethernet interface to conserver address space. In this case, enter the **ip unnumbered ethernet 0** command.
2. To create a string used to name the following list of authentication methods tried when a user logs in, refer to the **aaa authentication ppp** command. Authentication methods include RADIUS, TACACS+, and Kerberos.
3. To create an IP address pool, refer to the **ip local pool** global configuration command.

The "Group and Member Asynchronous Interface Examples" section later in this chapter contains an example of how to configure a group interface.

Verifying the Group Interface Configuration

To verify the group interface configuration and check if one of the asynchronous interfaces is up, use the **show interface async** command:

```
Router# show interface async 1

Async1 is up, line protocol is up
modem(slot/port)=1/0, csm_state(0x00000204)=CSM_IC4_CONNECTED, bchan_num=18
modem_status(0x0002): VDEV_STATUS_ACTIVE_CALL.

  Hardware is Async Serial
  Interface is unnumbered.  Using address of FastEthernet0 (10.1.1.10)
  MTU 1500 bytes, BW 115 Kbit, DLY 100000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive not set
  DTR is pulsed for 5 seconds on reset
  LCP Open
  Open: IPCP
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/5, 0 drops; input queue 1/5, 0 drops
  5 minute input rate 37000 bits/sec, 87 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     31063 packets input, 1459806 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     33 packets output, 1998 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

If you are having trouble, enter one of the following **debug** commands and then send a call into the access server. Interpret the output and make configuration changes accordingly.

- **undebug all**
- **debug ppp negotiation**
- **debug ppp authentication**
- **debug modem**
- **debug ip peer**

```
Router# undebug all
All possible debugging has been turned off
Router# debug ppp negotiation
PPP protocol negotiation debugging is on
Router# debug ppp authentication
PPP authentication debugging is on
Router# debug modem
Modem control/process activation debugging is on
Router# debug ip peer
IP peer address activity debugging is on
Router# show debug
General OS:
  Modem control/process activation debugging is on
Generic IP:
  IP peer address activity debugging is on
PPP:
  PPP authentication debugging is on
  PPP protocol negotiation debugging is on
Router#
*Mar  1 21:34:56.958: tty4: DSR came up
*Mar  1 21:34:56.962: tty4: Modem: IDLE->READY
*Mar  1 21:34:56.970: tty4: EXEC creation
*Mar  1 21:34:56.978: tty4: set timer type 10, 30 seconds
*Mar  1 21:34:59.722: tty4: Autoselect(2) sample 7E
*Mar  1 21:34:59.726: tty4: Autoselect(2) sample 7EFF
*Mar  1 21:34:59.730: tty4: Autoselect(2) sample 7EFF7D
*Mar  1 21:34:59.730: tty4: Autoselect(2) sample 7EFF7D23
*Mar  1 21:34:59.734: tty4 Autoselect cmd: ppp negotiate
*Mar  1 21:34:59.746: tty4: EXEC creation
*Mar  1 21:34:59.746: tty4: create timer type 1, 600 seconds
*Mar  1 21:34:59.786: ip_get_pool: As4: using pool default
*Mar  1 21:34:59.790: ip_get_pool: As4: returning address = 172.20.1.101
*Mar  1 21:34:59.794: tty4: destroy timer type 1 (OK)
*Mar  1 21:34:59.794: tty4: destroy timer type 0
*Mar  1 21:35:01.798: %LINK-3-UPDOWN: Interface Async4, changed state to up
*Mar  1 21:35:01.834: As4 PPP: Treating connection as a dedicated line
*Mar  1 21:35:01.838: As4 PPP: Phase is ESTABLISHING, Active Open
*Mar  1 21:35:01.842: As4 LCP: O CONFREQ [Closed] id 1 len 25
*Mar  1 21:35:01.846: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:01.850: As4 LCP:    AuthProto CHAP (0x0305C22305)
*Mar  1 21:35:01.854: As4 LCP:    MagicNumber 0x64E923A8 (0x050664E923A8)
*Mar  1 21:35:01.854: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:01.858: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:02.718: As4 LCP: I CONFREQ [REQsent] id 3 len 23
*Mar  1 21:35:02.722: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:02.726: As4 LCP:    MagicNumber 0x00472467 (0x050600472467)
*Mar  1 21:35:02.726: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:02.730: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:02.730: As4 LCP:    Callback 6  (0x0D0306)
*Mar  1 21:35:02.738: As4 LCP: O CONFREJ [REQsent] id 3 len 7
*Mar  1 21:35:02.738: As4 LCP:    Callback 6  (0x0D0306)
*Mar  1 21:35:02.850: As4 LCP: I CONFREQ [REQsent] id 4 len 20
```

```
*Mar  1 21:35:02.854: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:02.854: As4 LCP:    MagicNumber 0x00472467 (0x050600472467)
*Mar  1 21:35:02.858: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:02.858: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:02.862: As4 LCP: O CONFACK [REQsent] id 4 len 20
*Mar  1 21:35:02.866: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:02.870: As4 LCP:    MagicNumber 0x00472467 (0x050600472467)
*Mar  1 21:35:02.870: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:02.874: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:03.842: As4 LCP: TIMEout: State ACKsent
*Mar  1 21:35:03.842: As4 LCP: O CONFREQ [ACKsent] id 2 len 25
*Mar  1 21:35:03.846: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:03.850: As4 LCP:    AuthProto CHAP (0x0305C22305)
*Mar  1 21:35:03.854: As4 LCP:    MagicNumber 0x64E923A8 (0x050664E923A8)
*Mar  1 21:35:03.854: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:03.858: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:03.962: As4 LCP: I CONFACK [ACKsent] id 2 len 25
*Mar  1 21:35:03.966: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:03.966: As4 LCP:    AuthProto CHAP (0x0305C22305)
*Mar  1 21:35:03.970: As4 LCP:    MagicNumber 0x64E923A8 (0x050664E923A8)
*Mar  1 21:35:03.974: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:03.974: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:03.978: As4 LCP: State is Open
*Mar  1 21:35:03.978: As4 PPP: Phase is AUTHENTICATING, by this end
*Mar  1 21:35:03.982: As4 CHAP: O CHALLENGE id 1 len 26 from "nas-1"
*Mar  1 21:35:04.162: As4 CHAP: I RESPONSE id 1 len 26 from "krist"
*Mar  1 21:35:04.170: As4 AUTH: Started process 0 pid 47
*Mar  1 21:35:04.182: As4 CHAP: O SUCCESS id 1 len 4
*Mar  1 21:35:04.186: As4 PPP: Phase is UP
*Mar  1 21:35:04.190: As4 IPCP: O CONFREQ [Not negotiated] id 1 len 10
*Mar  1 21:35:04.194: As4 IPCP:    Address 172.20.1.2 (0x0306AC140102)
*Mar  1 21:35:04.202: As4 CDPCP: O CONFREQ [Closed] id 1 len 4
*Mar  1 21:35:04.282: As4 IPCP: I CONFREQ [REQsent] id 1 len 40
*Mar  1 21:35:04.282: As4 IPCP:    CompressType VJ 15 slots CompressSlotID (0x02
06002D0F01)
*Mar  1 21:35:04.286: As4 IPCP:    Address 0.0.0.0 (0x030600000000)
*Mar  1 21:35:04.290: As4 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
*Mar  1 21:35:04.294: As4 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
*Mar  1 21:35:04.298: As4 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
*Mar  1 21:35:04.302: As4 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
*Mar  1 21:35:04.306: As4 IPCP: O CONFREJ [REQsent] id 1 len 10
*Mar  1 21:35:04.310: As4 IPCP:    CompressType VJ 15 slots CompressSlotID (0x02
06002D0F01)
*Mar  1 21:35:04.314: As4 CCP: I CONFREQ [Not negotiated] id 1 len 15
*Mar  1 21:35:04.318: As4 CCP:    MS-PPC supported bits 0x00000001 (0x1206000000
01)
*Mar  1 21:35:04.318: As4 CCP:    Stacker history 1 check mode EXTENDED (0x11050
00104)
*Mar  1 21:35:04.322: As4 LCP: O PROTREJ [Open] id 3 len 21 protocol CCP
*Mar  1 21:35:04.326: As4 LCP:  (0x80FD0101000F12060000000111050001)
*Mar  1 21:35:04.330: As4 LCP:  (0x04)
*Mar  1 21:35:04.334: As4 IPCP: I CONFACK [REQsent] id 1 len 10
*Mar  1 21:35:04.338: As4 IPCP:    Address 172.20.1.2 (0x0306AC140102)
*Mar  1 21:35:04.342: As4 LCP: I PROTREJ [Open] id 5 len 10 protocol CDPCP (0x82
0701010004)
*Mar  1 21:35:04.342: As4 CDPCP: State is Closed
*Mar  1 21:35:05.186: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async4, ch
anged state to up
*Mar  1 21:35:05.190: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:35:05.190: As4 PPP: Trying to negotiate NCP for Link cdp
*Mar  1 21:35:05.194: As4 CDPCP: State is Closed
*Mar  1 21:35:05.198: As4 CDPCP: TIMEout: State Closed
*Mar  1 21:35:05.202: As4 CDPCP: State is Listen
*Mar  1 21:35:06.202: As4 IPCP: TIMEout: State ACKrcvd
```

```
*Mar  1 21:35:06.206: As4 IPCP: O CONFREQ [ACKrcvd] id 2 len 10
*Mar  1 21:35:06.206: As4 IPCP:    Address 172.20.1.2 (0x0306AC140102)
*Mar  1 21:35:06.314: As4 IPCP: I CONFACK [REQsent] id 2 len 10
*Mar  1 21:35:06.318: As4 IPCP:    Address 172.20.1.2 (0x0306AC140102)
*Mar  1 21:35:07.274: As4 IPCP: I CONFREQ [ACKrcvd] id 2 len 34
*Mar  1 21:35:07.278: As4 IPCP:    Address 0.0.0.0 (0x030600000000)
*Mar  1 21:35:07.282: As4 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
*Mar  1 21:35:07.286: As4 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
*Mar  1 21:35:07.286: As4 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
*Mar  1 21:35:07.290: As4 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
*Mar  1 21:35:07.294: As4 IPCP: O CONFNAK [ACKrcvd] id 2 len 34
*Mar  1 21:35:07.298: As4 IPCP:    Address 172.20.1.101 (0x0306AC140165)
*Mar  1 21:35:07.302: As4 IPCP:    PrimaryDNS 172.20.5.100 (0x8106AC140564)
*Mar  1 21:35:07.306: As4 IPCP:    PrimaryWINS 172.20.5.101 (0x8206AC140565)
*Mar  1 21:35:07.310: As4 IPCP:    SecondaryDNS 172.20.6.100 (0x8306AC140664)
*Mar  1 21:35:07.314: As4 IPCP:    SecondaryWINS 172.20.6.101 (0x8406AC140665)
*Mar  1 21:35:07.426: As4 IPCP: I CONFREQ [ACKrcvd] id 3 len 34
*Mar  1 21:35:07.430: As4 IPCP:    Address 172.20.1.101 (0x0306AC140165)
*Mar  1 21:35:07.434: As4 IPCP:    PrimaryDNS 172.20.5.100 (0x8106AC140564)
*Mar  1 21:35:07.438: As4 IPCP:    PrimaryWINS 172.20.5.101 (0x8206AC140565)
*Mar  1 21:35:07.442: As4 IPCP:    SecondaryDNS 172.20.6.100 (0x8306AC140664)
*Mar  1 21:35:07.446: As4 IPCP:    SecondaryWINS 172.20.6.101 (0x8406AC140665)
*Mar  1 21:35:07.446: ip_get_pool: As4: validate address = 172.20.1.101
*Mar  1 21:35:07.450: ip_get_pool: As4: using pool default
*Mar  1 21:35:07.450: ip_get_pool: As4: returning address = 172.20.1.101
*Mar  1 21:35:07.454: set_ip_peer_addr: As4: address = 172.20.1.101 (3) is redun
dant
*Mar  1 21:35:07.458: As4 IPCP: O CONFACK [ACKrcvd] id 3 len 34
*Mar  1 21:35:07.462: As4 IPCP:    Address 172.20.1.101 (0x0306AC140165)
*Mar  1 21:35:07.466: As4 IPCP:    PrimaryDNS 172.20.5.100 (0x8106AC140564)
*Mar  1 21:35:07.470: As4 IPCP:    PrimaryWINS 172.20.5.101 (0x8206AC140565)
*Mar  1 21:35:07.474: As4 IPCP:    SecondaryDNS 172.20.6.100 (0x8306AC140664)
*Mar  1 21:35:07.474: As4 IPCP:    SecondaryWINS 172.20.6.101 (0x8406AC140665)
*Mar  1 21:35:07.478: As4 IPCP: State is Open
*Mar  1 21:35:07.490: As4 IPCP: Install route to 172.20.1.101
*Mar  1 21:35:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:36:12.614: tty0: timer type 1 expired
*Mar  1 21:36:12.614: tty0:  Exec timer (continued)
*Mar  1 21:36:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:37:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:38:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:39:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:40:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:41:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:42:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:43:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
```

Configuring Asynchronous Rotary Line Queueing

The Cisco IOS Asynchronous Rotary Line Queueing feature allows Telnet connection requests to busy asynchronous rotary groups to be queued so that users automatically obtain the next available line, rather than needing to try repeatedly to open a Telnet connection. The Cisco IOS software sends a periodic message to the user to update progress in the connection queue.

This feature allows users to make effective use of the asynchronous rotary groups on a Cisco router to access legacy mainframes or other serial devices with a limited number of asynchronous ports that might be used by a large number of users. Users that are unable to make a Telnet connection on the first attempt are assured of eventual success in an orderly process. They are no longer required to guess when a line might be available and to retry manually again and again.

Connections are authenticated using the method specified for the line configurations for the asynchronous rotary group. If a connection is queued, authentication is done prior to queueing and no authentication is done when the connection is later established.

Make sure you comply with the following requirements when configuring asynchronous rotary line queueing:

- Configure more virtual terminal lines than will ever be used by waiting asynchronous rotary connection attempts. Even when the queue is at its maximum, there must be at least one virtual terminal line available so that system operators or network administrators can use Telnet to access the router to show, debug, or configure system performance.
- When adding lines to a rotary group, all lines must be either queued or not queued. A mixture of queued and unenqueued lines in the same rotary group is not supported and can result in unexpected behavior.
- All lines within a queued rotary group need to use the same authentication method. Using different authentication methods within the same rotary group can result in unexpected behavior.

To configure asynchronous rotary line queueing, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router (config)# line [aux | console | tty | vty] line-number [ending-line-number]`	Starts line configuration mode on the line type and numbers specified.
Step 2	`Router(config-line)# rotary group [queued | round-robin]`	Enables asynchronous rotary line queueing on the designated line or group of lines. The optional **round-robin** keyword selects a round-robin port selection algorithm instead of the default (**queued**) linear port selection algorithm.

See the "Rotary Group Examples" section for configuration examples.

Verifying Asynchronous Rotary Line Queueing

To verify operation of asynchronous rotary line queueing, perform the following tasks:

- Use the **show line** command in EXEC mode to check the status of the vty lines.
- Use the **show line async-queue** command in EXEC mode to check the status of queued connection requests.

Troubleshooting Asynchronous Rotary Lines

If asynchronous rotary line queueing is not operating correctly, use the following **debug** commands in privileged EXEC mode to determine where the problem may lie:

- **debug async async-queue**
- **debug ip tcp transactions**
- **debug modem**

Refer to the *Cisco IOS Debug Command Reference* for information about these commands.

Monitoring and Maintaining Asynchronous Rotary Line Queues

To display queued lines and to remove lines from the queue, use the following commands in EXEC mode as needed:

Command	Purpose
`Router# show line async-queue rotary-group`	Displays which lines are queued.
`Router# clear line async-queue rotary-group`	Clears all rotary queues or the specified rotary queue. If the *rotary-group* argument is not specified, all rotary queues are removed.

Configuring Autoselect

Autoselect is used by the access server to sense the protocol being received on an incoming line and to launch the appropriate protocol. Autoselect can be used for AppleTalk Remote Access (ARA), PPP, or SLIP.

When using Autoselect, "login" authentication is bypassed, so if security is required, it must be performed at the protocol level, that is, the AppleTalk Remote Access Protocol (ARAP) or PPP authentication. SLIP does not offer protocol layer authentication.

To configure the Cisco IOS software to allow an ARA, PPP, or SLIP session to start automatically, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# autoselect {arap | ppp | slip | during login}`	Configures a line to automatically start an ARA, PPP, or SLIP session.

The **autoselect** command enables the Cisco IOS software to start a process automatically when a start character is received.

The **autoselect** command bypasses the login prompt and enables the specified session to begin automatically. However, when the **autoselect** command is entered with the **during login** keyword, the username or password prompt appears without the need to press the Return key; thus "login" users will get a prompt right away without needing to press the Return key. While the username or password prompt is displayed, you can choose either to answer these prompts or to send packets from an autoselected protocol.

Normally a router avoids line and modem noise by clearing the initial data received within the first one or two seconds. However, when the autoselect PPP feature is configured, the router flushes characters initially received and then waits for more traffic. This flush causes timeout problems with applications that send only one carriage return. To ensure that the input data sent by a modem or other asynchronous device is not lost after line activation, enter the **flush-at-activation** line configuration command.

Note When the **autoselect** command is used, the activation character should be set to the default Return, and exec-character-bits should be set to 7. If you change these defaults, the application cannot recognize the activation request.

See the "High-Density Dial-In Solution Using Autoselect and EXEC Control Example" section for an example that makes use of the autoselect feature.

Verifying Autoselect PPP

The following trace appears when the **debug modem** and **debug ppp negotiation** commands are enabled. As PPP calls pass through the access server, you should see this output.

When autoselect is used, "login" authentication is bypassed. If security is required, it must be performed at the protocol level (that is, ARAP or PPP authentication). SLIP does not offer protocol layer authentication.

```
22:21:02: TTY1: DSR came up
22:21:02: tty1: Modem: IDLE->READY
22:21:02: TTY1: Autoselect started
22:21:05: TTY1: Autoselect sample 7E
22:21:05: TTY1: Autoselect sample 7EFF
22:21:05: TTY1: Autoselect sample 7EFF7D
22:21:05: TTY1 Autoselect cmd: ppp default
22:21:05: TTY1: EXEC creation
%LINK-3-UPDOWN: Interface Async1, changed state to up
22:21:07: ppp: sending CONFREQ, type = 2 (CI_ASYNCMAP), value = A0000
22:21:07: ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 23BE13AA
22:21:08: PPP Async1: state = REQSENT fsm_rconfack(0xC021): rcvd id 0x11
22:21:08: ppp: config ACK received, type = 2 (CI_ASYNCMAP), value = A0000
22:21:08: ppp: config ACK received, type = 5 (CI_MAGICNUMBER), value = 23BE13AA
22:21:08: ppp: config ACK received, type = 7 (CI_PCOMPRESSION)
22:21:08: ppp: config ACK received, type = 8 (CI_ACCOMPRESSION)
22:21:08: PPP Async1: received config for type = 0x2 (ASYNCMAP) value = 0x0 acked
22:21:08: PPP Async1: received config for type = 0x5 (MAGICNUMBER) value = 0x2A acked
22:21:08: PPP Async1: received config for type = 0x7 (PCOMPRESSION) acked
22:21:08: PPP Async1: received config for type = 0x8 (ACCOMPRESSION) acked
22:21:08: ipcp: sending CONFREQ, type = 3 (CI_ADDRESS), Address = 172.16.1.1
22:21:08: ppp Async1: ipcp_reqci: rcvd COMPRESSTYPE (rejected) (REJ)
22:21:08: ppp Async1: Negotiate IP address: her address 0.0.0.0 (NAK with address
172.16.1.100) (NAK)
22:21:08: ppp: ipcp_reqci: returning CONFREJ.
22:21:08: PPP Async1: state = REQSENT fsm_rconfack(0x8021): rcvd id 0x9
22:21:08: ipcp: config ACK received, type = 3 (CI_ADDRESS), Address = 172.16.1.1
22:21:08: ppp Async1: Negotiate IP address: her address 0.0.0.0 (NAK with address
172.16.1.100) (NAK)
22:21:08: ppp: ipcp_reqci: returning CONFNAK.
22:21:09: ppp Async1: Negotiate IP address: her address 172.16.1.100 (ACK)
22:21:09: ppp: ipcp_reqci: returning CONFACK.
%LINEPROTO-5-UPDOWN: Line protocol on Interface Async1, changed state to up
```

Verifying Autoselect ARA

The following trace appears when the **debug modem** and **debug arap internal** commands are enabled. As ARA version 2.0 calls pass through the access server, this output is displayed.

```
20:45:11: TTY3: DSR came up
20:45:11: tty3: Modem: IDLE->READY
20:45:11: TTY3: EXEC creation
20:45:11: TTY3: Autoselect(2) sample 1
20:45:11: TTY3: Autoselect(2) sample 11B
20:45:12: TTY3: Autoselect(2) sample 11B02
20:45:18: ARAP: ---------- SRVRVERSION ----------
20:45:19: ARAP: ---------- ACKing 0 ----------
20:45:19: ARAP: ---------- AUTH_CHALLENGE ----------
20:45:21: ARAP: ---------- ACKing 1 ----------
20:45:21: ARAP: ---------- AUTH_RESPONSE ----------
20:45:21: ARAP: ---------- STARTINFOFROMSERVER ----------
20:45:22: ARAP: ---------- ACKing 2 ----------
22:45:22: ARAP: ---------- ZONELISTINFO ----------
```

```
22:45:22: ARAP: ---------- ZONELISTINFO ----------
22:45:22: ARAP: ---------- ZONELISTINFO ----------
```

The following trace is for ARA version 1.0 calls:

```
22:31:45: TTY1: DSR came up
22:31:45: tty1: Modem: IDLE->READY
22:31:45: TTY1: Autoselect started
22:31:46: TTY1: Autoselect sample 16
22:31:46: TTY1: Autoselect sample 1610
22:31:46: TTY1: Autoselect sample 161002
22:31:47: ARAP: ---------- SRVRVERSION ----------
22:31:47: ARAP: ---------- ACKing 0 ----------
22:31:47: ARAP: ---------- AUTH_CHALLENGE ----------
22:31:47: ARAP: ---------- ACKing 1 ----------
22:31:47: ARAP: ---------- AUTH_RESPONSE ----------
22:31:47: ARAP: ---------- STARTINFOFROMSERVER ----------
22:31:48: ARAP: ---------- ACKing 2 ----------
22:31:48: ARAP: ---------- ZONELISTINFO ----------
22:31:48: ARAP: ---------- ZONELISTINFO ----------
22:31:49: ARAP: ---------- ZONELISTINFO ----------
```

How to Configure Other Asynchronous Line and Interface Features

This section describes the following asynchronous line and interface configurations:

- Configuring the Auxiliary (AUX) Port
- Establishing and Controlling the EXEC Process
- Enabling Routing on Asynchronous Interfaces
- Configuring Dedicated or Interactive PPP and SLIP Sessions
- Conserving Network Addresses
- Using Advanced Addressing Methods for Remote Devices
- Optimizing Available Bandwidth

Configuring the Auxiliary (AUX) Port

The AUX (auxiliary) port is typically configured as an asynchronous serial interface on routers without built-in asynchronous interfaces. To configure the AUX port as an asynchronous interface, configure it first as an auxiliary line with the **line aux 1** global configuration command.

The AUX port sends a data terminal ready (DTR) signal only when a Telnet connection is established. The auxiliary port does not use request to send/clear to send (RTS/CTS) handshaking for flow control. To understand the differences between standard asynchronous interfaces and AUX ports configured as an asynchronous interface, refer to Table 4. To enable the auxiliary port, use the following command in global configuration mode:

Command	Purpose
`Router(config)# line aux line-number`	Enables the auxiliary serial DTE port.

You cannot use the auxiliary (AUX) port as a second console port. To use the AUX port as a console port, you must order a special cable from your technical support personnel.

On an access server, you can configure any of the available asynchronous interfaces (1 through 8, 16, or 48). The auxiliary port (labeled AUX on the back of the product) can also be configured as an asynchronous serial interface, although performance on the AUX port is much slower than on standard asynchronous interfaces and the port does not support some features.

Table 4 illustrates why asynchronous interfaces permit substantially better performance than AUX ports configured as asynchronous interfaces.

Table 4 Differences Between the Asynchronous Port and the Auxiliary (AUX) Port

Feature	Asynchronous Interface	Auxiliary Port
Maximum speed	115200 bps	38400 bps
DMA buffering support[1]	Yes	No
PPP framing on chip[2]	Yes	No
IP fast switching[3]	Yes	No

1. Direct Memory Access (DMA) buffering moves data packets directly to and from system memory without interrupting the main CPU. This process removes overhead from the CPU and increases overall system performance.
2. PPP framing on a hardware chip removes overhead from the CPU on the router, which enables the router to sustain 115200 bps throughput on all asynchronous ports simultaneously.
3. After the destination of the first IP packet is added to the fast switching cache, it is fast switched to and from other interfaces with minimal involvement from the main processor.

On routers without built-in asynchronous interfaces, only the AUX port can be configured as an asynchronous serial interface. To configure the AUX port as an asynchronous interface, you must also configure it as an auxiliary line with the **line aux 1** command. Access servers do not have this restriction. Use the line command with the appropriate line configuration commands for modem control, such as speed.

Only IP packets can be sent across lines configured for SLIP. PPP supports transmission of IP, Internet Packet Exchange (IPX), and AppleTalk packets on an asynchronous serial interface.

See the "Line AUX Configuration Example" section for an example that shows how to configure the AUX port.

Establishing and Controlling the EXEC Process

By default, the Cisco IOS software starts an EXEC process on all lines. However, you can control EXEC processes, as follows:

- Turn the EXEC process on or off. (A serial printer, for example, should not have an EXEC session started.)
- Set the idle terminal timeout interval.

The EXEC command interpreter waits for a specified amount of time to receive user input. If no input is detected, the EXEC facility resumes the current connection. If no connections exist, it returns the terminal to the idle state and disconnects the incoming connection.

To control the EXEC process, use the following commands in line configuration mode:

	Command	Purpose
Step 1	`Router(config-line)# exec`	Turns on EXEC processes.
Step 2	`Router(config-line)# exec-timeout` *minutes* [*seconds*]	Sets the idle terminal timeout interval.

See the "High-Density Dial-In Solution Using Autoselect and EXEC Control Example" section for an example of configuring control over the EXEC process.

Enabling Routing on Asynchronous Interfaces

To route IP packets on an asynchronous interface, use one of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# async dynamic routing`	Configures an asynchronous interface for dynamic routing. Use this command to manually bring up PPP from an EXEC session.
`Router(config-if)# async default routing`	Automatically configures an asynchronous interface for routing. Use this command to enable two routers to communicate over an asynchronous dial backup link.

The **async dynamic routing** command routes IP packets on an asynchronous interface, which permits you to enable the Interior Gateway Routing Protocol (IGRP), Routing Information Protocol (RIP), and Open Shortest Path First (OSPF) routing protocols for use when the user makes a connection using the **ppp** or **slip** EXEC commands. The user must, however, specify the **/routing** keyword at the SLIP or PPP command line.

For asynchronous interfaces in interactive mode, the **async default routing** command causes the **ppp** and **slip** EXEC commands to be interpreted as though the **/route** switch had been included in the command. For asynchronous interfaces in dedicated mode, the **async dynamic routing** command enables routing protocols to be used on the line. Without the **async default routing** command, there is no way to enable the use of routing protocols automatically on a dedicated asynchronous interface.

See the following sections for examples of enabling routing on asynchronous interfaces:

- Asynchronous Interface As the Only Network Interface Example
- IGRP Configuration Example

Configuring Dedicated or Interactive PPP and SLIP Sessions

You can configure one or more asynchronous interfaces on your access server (and one on a router) to be in dedicated network interface mode. In dedicated mode, an interface is automatically configured for SLIP or PPP connections. There is no user prompt or EXEC level, and no end-user commands are required to initiate remote-node connections. If you want a line to be used only for SLIP or PPP connections, configure the line for dedicated mode.

In interactive mode, a line can be used to make any type of connection, depending on the EXEC command entered by the user. For example, depending on its configuration, the line could be used for Telnet or XRemote connections, or SLIP or PPP encapsulation. The user is prompted for an EXEC command before a connection is initiated.

You can configure an asynchronous interface to be in dedicated network mode. When the interface is configured for dedicated mode, the end user cannot change the encapsulation method, address, or other parameters.

To configure an interface for dedicated network mode or to return it to interactive mode, use one of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# async mode dedicated`	Places the line into dedicated asynchronous network mode.
`Router(config-if)# async mode interactive`	Returns the line to interactive mode.

By default, no asynchronous mode is configured. In this state, the line is not available for inbound networking because the SLIP and PPP connections are disabled.

See the "Dedicated Asynchronous Interface Configuration Example" section for an example of how to configure a dedicated asynchronous interface.

Conserving Network Addresses

When asynchronous routing is enabled, you might need to conserve network addresses by configuring the asynchronous interfaces as *unnumbered*. An unnumbered interface does not have an address. Network resources are therefore conserved because fewer network numbers are used and routing tables are smaller.

To configure an unnumbered interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip unnumbered` *type number*	Conserves IP addresses by configuring the asynchronous interfaces as unnumbered, and assigns the IP address of the interface type that you want to leverage.

Whenever the unnumbered interface generates a packet (for example, a routing update), it uses the address of the specified interface as the source address of the IP packet. It also uses the address of the specified interface to determine which routing processes are sending updates over the unnumbered interface.

You can use the IP unnumbered feature even if the system on the other end of the asynchronous link does not support it. The IP unnumbered feature is transparent to the other end of the link because each system bases its routing activities on information in the routing updates it receives and on its own interface address.

See the "Network Address Conservation Using the ip unnumbered Command Example" section for an example of how to conserve network addresses.

Using Advanced Addressing Methods for Remote Devices

You can control whether addressing is dynamic (the user specifies the address at the EXEC level when making the connection) or whether default addressing is used (the address is forced by the system). If you specify dynamic addressing, the router must be in interactive mode and the user will enter the address at the EXEC level.

It is common to configure an asynchronous interface to have a default address and to allow dynamic addressing. With this configuration, the choice between the default address or dynamic addressing is made by the users when they enter the **slip** or **ppp** EXEC command. If the user enters an address, it is used, and if the user enters the **default** keyword, the default address is used.

This section describes the following optional tasks:

- Assigning a Default Asynchronous Address
- Allowing an Asynchronous Address to Be Assigned Dynamically

Assigning a Default Asynchronous Address

To assign a permanent default asynchronous address, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# peer default ip address` *ip-address*	Assigns a default IP address to an asynchronous interface.

Use the **no** form of this command to disable the default address. If the server has been configured to authenticate asynchronous connections, you are prompted for a password after you enter the **slip default** or **ppp default** EXEC command before the line is placed into asynchronous mode.

The assigned default address is implemented when the user enters the **slip default** or **ppp default** EXEC command. The transaction is validated by the TACACS server, when enabled, and the line is put into network mode using the address that is in the configuration file.

Configuring a default address is useful when the user is not required to know the IP address to gain access to a system (for example, users of a server that is available to many students on a campus). Instead of each user being required to know an IP address, they only need to enter the **slip default** or **ppp default** EXEC command and let the server select the address to use.

See the section "Making Additional Remote Node Connections" in the chapter "Configuring Asynchronous SLIP and PPP" in this publication for more information about the **slip** and **ppp** EXEC commands.

See the following sections for examples:

- Modem Asynchronous Group Example
- Configuring Specific IP Addresses for an Interface
- IP and PPP Asynchronous Interface Configuration Example

Allowing an Asynchronous Address to Be Assigned Dynamically

When a line is configured for dynamic assignment of asynchronous addresses, the user enters the **slip** or **ppp** EXEC command and is prompted for an address or logical host name. The address is validated by TACACS, when enabled, and the line is assigned the given address and put into asynchronous mode.

Assigning asynchronous addresses dynamically is useful when you want to assign set addresses to users. For example, an application on a personal computer that automatically dials in using Serial Line Internet Protocol (SLIP) and polls for electronic mail messages can be set up to dial in periodically and enter the required IP address and password.

To assign asynchronous addresses dynamically, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# async dynamic address`	Allows the IP address to be assigned when the protocol is initiated.

The dynamic addressing features of the internetwork allow packets to get to their destination and back regardless of the access server, router, or network they are sent from. For example, if a host such as a laptop computer moves from place to place, it can keep the same address no matter where it is dialing in from.

Logical host names are first converted to uppercase and then sent to the TACACS server for authentication.

See the following sections for examples of configurations that allow asynchronous addresses to be assigned dynamically:

- Access Restriction on the Asynchronous Interface Example
- Asynchronous Routing and Dynamic Addressing Configuration Example
- Network Address Conservation Using the ip unnumbered Command Example

Optimizing Available Bandwidth

Asynchronous lines have relatively low bandwidth and can easily be overloaded, resulting in slow traffic across these lines.

To optimize available bandwidth, perform either of the following optional tasks:

- Configuring Header Compression
- Forcing Header Compression at the EXEC Level

Configuring Header Compression

One way to optimize available bandwidth is by using TCP header compression. Van Jacobson TCP header compression (defined by RFC 1144) can increase bandwidth availability two- to five-fold when compared to lines not using header compression. Theoretically, it can improve bandwidth availability by a ratio of seven to one.

To configure header compression, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip tcp header-compression [on | off | passive]`	Configures Van Jacobson TCP header compression on the asynchronous link.

Forcing Header Compression at the EXEC Level

On SLIP interfaces, you can force header compression at the EXEC prompt on a line on which header compression has been set to passive. This option allows more efficient use of the available bandwidth and does not require entering privileged configuration mode.

To implement header compression, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip tcp header-compression passive`	Allows status of header compression to be assigned at the user level.

For PPP interfaces, the **passive** option functions the same as the **on** option.

See the following sections for examples of header compression:

- TCP Header Compression Configuration Example
- Network Address Conservation Using the ip unnumbered Command Example
- IGRP Configuration Example

Configuration Examples for Asynchronous Interfaces and Lines

This section provides the following asynchronous interface configuration examples:

- Interface and Line Configuration Examples
- Line AUX Configuration Example
- Rotary Group Examples
- Dedicated Asynchronous Interface Configuration Example
- Access Restriction on the Asynchronous Interface Example
- Group and Member Asynchronous Interface Examples
- Asynchronous Interface Address Pool Examples
- IP and SLIP Using an Asynchronous Interface Example
- IP and PPP Asynchronous Interface Configuration Example
- Asynchronous Routing and Dynamic Addressing Configuration Example
- TCP Header Compression Configuration Example
- Network Address Conservation Using the ip unnumbered Command Example
- Asynchronous Interface As the Only Network Interface Example
- Routing on a Dedicated Dial-In Router Example
- IGRP Configuration Example

Interface and Line Configuration Examples

This section contains the following examples:

- Asynchronous Interface Backup DDR Configuration Example
- Passive Header Compression and Default Address Example
- High-Density Dial-In Solution Using Autoselect and EXEC Control Example
- Asynchronous Line Backup DDR Configuration Example

Asynchronous Interface Backup DDR Configuration Example

The following is an example of one asynchronous interface configuration on a Cisco AS2511-RJ access server that is used in an asynchronous backup DDR scenario:

```
interface async 1
 description ASYNC LINE 5293731 TO HIGHWAY
 encapsulation ppp
 async default routing
 async mode dedicated
 dialer in-band
 dialer map ip 192.168.10.2 name Router2 broadcast
 dialer-group 1
 ppp authentication chap
```

Passive Header Compression and Default Address Example

The following configuration shows interface and line configuration. The interface is configured with access lists, passive header compression, and a default address. The line is configured for TACACS authentication.

```
interface async 1
 ip access-group 1 in
 ip access-group 1 out
 ip tcp header-compression passive
 async default ip address 172.31.176.201

line 1
 login tacacs
 location 457-5xxx
 exec-timeout 20 0
 password XXXXXXXX
 session-timeout 20
 stopbits 1
```

High-Density Dial-In Solution Using Autoselect and EXEC Control Example

The following example configures a Cisco AS5800 access server, which is used as a high-density dial-in solution:

```
line 1/2/00 1/9/71
 session-timeout 30
 exec-timeout 30 0
 absolute-timeout 240
 autoselect during-login
 autoselect ppp
```

```
 modem InOut
 transport preferred none
 transport input all
```

Asynchronous Line Backup DDR Configuration Example

The following example configures one asynchronous line on a Cisco AS2511-RJ access server that is used in an asynchronous backup DDR scenario:

```
line 1
 modem InOut
 speed 115200
 transport input all
 flowcontrol hardware
```

Line AUX Configuration Example

In the following example, the asynchronous interface corresponds to the AUX port. Use the **show line** command to determine which asynchronous interface corresponds to the AUX port. The IP address on the AUX ports of both routers are in the same subnet

```
interface Async1
 ip address 192.168.10.1 255.255.255.0
 encapsulation ppp
 async dynamic routing
 async mode dedicated
!
no ip classless
ip route 0.0.0.0 0.0.0.0 Async1 /Default route points to the Async1 (AUX port) interface.
!
!
logging buffered
!
line con 0
 exec-timeout 0 0
line aux 0
 modem InOut
 transport input all
 rxspeed 38400
 txspeed 38400
```

Rotary Group Examples

The following example establishes a rotary group consisting of virtual terminal lines 2 through 4 and defines a password on those lines. By using Telnet to connect to TCP port 3001, the user gets the next free line in the rotary group. The user need not remember the range of line numbers associated with the password.

```
line vty 2 4
 rotary 1
 password letmein
 login
```

The following example enables asynchronous rotary line queueing:

```
line 1 2
 rotary 1 queued
```

The following example enables asynchronous rotary line queueing using the round-robin algorithm:

```
line 1 2
 rotary 1 queued round-robin
```

Dedicated Asynchronous Interface Configuration Example

The following example shows how to assign an IP address to an asynchronous interface and place the line in dedicated network mode. Setting the stop bit to 1 is a performance enhancement.

```
line 20
 location Department PC Lab
 stopbits 1
 speed 19200
!
interface async 20
 async default ip address 172.18.7.51
 async mode dedicated
```

Access Restriction on the Asynchronous Interface Example

The following example shows how to allow most terminal users access to anything on the local network, but restrict access to certain servers designated as asynchronous servers:

```
! access list for normal connections
access-list 1 permit 192.168.0.0 0.0.255.255
!
access-list 2 permit 192.168.42.55
access-list 2 permit 192.168.111.1
access-list 2 permit 192.168.55.99
!
line 1
 speed 19200
 flow hardware
 modem inout
interface async 1
 async mode interactive
 async dynamic address
 ip access-group 1 out
 ip access-group 2 in
```

Group and Member Asynchronous Interface Examples

The following examples are included in this section:

- Asynchronous Group Interface Examples
- Modem Asynchronous Group Example
- High-Density Dial-In Solution Using an Asynchronous Group

Asynchronous Group Interface Examples

The following example shows how to create an asynchronous group interface 0 with group interface members 2 through 7, beginning in global configuration mode:

```
interface group-async 0
 group-range 2 7
```

The following example shows how you need to configure asynchronous interfaces 1, 2, and 3 separately if you do not have a group interface configured:

```
interface Async1
 ip unnumbered Ethernet0
 encapsulation ppp
 async default ip address 172.30.1.1
 async mode interactive
 async dynamic routing
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 async default ip address 172.30.1.2
 async mode interactive
 async dynamic routing
!
interface Async3
 ip unnumbered Ethernet0
!
 encapsulation ppp
 async default ip address 172.30.1.3
 async mode interactive
 async dynamic routing
```

The following example configures the same interfaces, but from a single group asynchronous interface:

```
interface Group-Async 0
 ip unnumbered Ethernet0
 encapsulation ppp
 async mode interactive
 async dynamic routing
 group-range 1 3
 member 1 async default ip address 172.30.1.1
 member 2 async default ip address 172.30.1.2
 member 3 async default ip address 172.30.1.3
```

Modem Asynchronous Group Example

To configure a group asynchronous interface, specify the group async number (an arbitrary number) and the group range (beginning and ending asynchronous interface number).

The following example shows the process of creating and configuring a group asynchronous interface for asynchronous interfaces 1 through 96 on a Cisco AS5300 access server, which is loaded with ninety-six 56K MICA technologies modems:

```
interface group-async 1
 ip unnumbered ethernet 0
 encapsulation ppp
 async mode interactive
 ppp authentication chap pap
 peer default ip address pool default
 group-range 1 96
```

High-Density Dial-In Solution Using an Asynchronous Group

The following example configures a Cisco AS5800 access server that is used as a high-density dial-in solution:

```
interface group-async 0
 ip unnumbered FastEthernet0/2/0
 encapsulation ppp
 async mode interactive
 peer default ip address pool default
 no cdp enable
 ppp authentication chap
 hold-queue 10 in
 group-range 1/2/00 1/9/71
```

Asynchronous Interface Address Pool Examples

The following sections provide examples of the use of Dynamic Host Configuration Protocol (DHCP) and local pooling mechanisms:

- DHCP Pooling Example
- Local Pooling Example
- Configuring Specific IP Addresses for an Interface

DHCP Pooling Example

The following global configuration example enables DHCP proxy-client status on all asynchronous interfaces on the access server:

```
ip address-pool dhcp-proxy-client
```

The following global configuration example shows how to specify which DHCP servers are used on your network. You can specify up to four servers using IP addresses or names. If you do not specify servers, the default is to use the IP limited broadcast address of 255.255.255.255 for transactions with any and all discovered DHCP servers.

```
ip dhcp-server jones smith wesson
```

The following interface configuration example illustrates how to disable DHCP proxy-client functionality on asynchronous interface 1:

```
async interface
interface 1
 no peer default ip address
```

Local Pooling Example

The following example shows how to select the IP pooling mechanism and how to create a pool of local IP addresses that are used when a client dials in on an asynchronous line. The default address pool comprises IP addresses 172.30.0.1 through 172.30.0.28.

```
! This command tells the access server to use a local pool.
```

```
ip address-pool local
! This command defines the ip address pool.
! The address pool is named group1 and comprised of addresses.
! 172.30.0.1 through 172.30.0.28 inclusive
ip local-pool group1 172.30.0.1 172.30.0.28
```

Configuring Specific IP Addresses for an Interface

The following example shows how to configure the access server so that it will use the default address pool on all interfaces except interface 7, on which it will use an address pool called lass:

```
ip address-pool local
ip local-pool lass 172.30.0.1
 async interface
 interface 7
 peer default ip address lass
```

IP and SLIP Using an Asynchronous Interface Example

The following example configures IP and SLIP on asynchronous interface 6. The IP address for the interface is assigned to Ethernet 0, interactive mode has been enabled, and the IP address of the client PC running SLIP has been specified.

IP and the appropriate IP routing protocols have already been enabled on the access server or router.

```
interface async 6
 ip unnumbered ethernet 0
 encapsulation slip
 async mode interactive
 async default ip address 172.18.1.128
```

IP and PPP Asynchronous Interface Configuration Example

The following example configures IP and PPP on asynchronous interface 6. The IP address for the interface is assigned to Ethernet 0, interactive mode has been enabled, and the IP address of the client PC running PPP has been specified. IP and the appropriate IP routing protocols have already been enabled on the access server or router.

```
interface async 6
 ip unnumbered ethernet 0
 encapsulation ppp
 async mode interactive
 peer default ip address 172.18.1.128
```

Asynchronous Routing and Dynamic Addressing Configuration Example

The following example shows a simple configuration that allows routing and dynamic addressing.

With this configuration, if the user specifies **/routing** in the EXEC **slip** or **ppp** command, routing protocols will be sent and received.

```
interface async 6
 async dynamic routing
 async dynamic address
```

TCP Header Compression Configuration Example

The following example configures asynchronous interface 7 with a default IP address, allowing header compression if it is specified in the **slip** or **ppp** connection command entered by the user or if the connecting system sends compressed packets.

```
interface async 7
 ip address 172.31.79.1
 async default ip address 172.31.79.2
 ip tcp header-compression passive
```

Network Address Conservation Using the ip unnumbered Command Example

The following example shows how to configure your router for routing using unnumbered interfaces. The source (local) address is shared between the Ethernet 0 and asynchronous 6 interfaces (172.18.1.1). The default remote address is 172.18.1.2.

```
interface ethernet 0
 ip address 172.18.1.1 255.255.255.0
!
interface async 6
 ip unnumbered ethernet 0
 async dynamic routing
! Default address is on the local subnet.
 async dynamic address
 async default ip address 172.18.1.2
 ip tcp header-compression passive
```

The following example shows how the IP unnumbered configuration works. Although the user is assigned an address, the system response shows the interface as unnumbered, and the address entered by the user will be used only in response to BOOTP requests.

```
Router> slip /compressed 10.11.11.254
Password:
Entering async mode.
Interface IP address is unnumbered, MTU is 1500 bytes.
Header compression is On.
```

Asynchronous Interface As the Only Network Interface Example

The following example shows how one of the asynchronous lines can be used as the only network interface. The router is used primarily as a terminal server, but is at a remote location and dials in to the central site for its only network connection.

```
ip default-gateway 10.11.12.2
interface ethernet 0
 shutdown
interface async 1
 async dynamic routing
 ip tcp header-compression on
 async default ip address 10.11.16.12
 async mode dedicated
 ip address 10.11.12.32 255.255.255.0
```

Routing on a Dedicated Dial-In Router Example

The following example shows how a router is set up as a dedicated dial-in router. Interfaces are configured as IP unnumbered to conserve network resources, primarily IP addresses.

```
ip routing
interface ethernet 0
 ip address 10.129.128.2 255.255.255.0
!
interface async 1
 ip unnumbered ethernet 0
 async dynamic routing
! The addresses assigned with SLIP or PPP EXEC commands are not used except
! to reply to BOOTP requests.
! Normally, the routers dialing in will have their own address and not use BOOTP at all.
 async default ip address 10.11.11.254
!
interface async 2
 ip unnumbered ethernet 0
 async default ip address 10.11.12.16
 ip tcp header-compression passive
 async mode dedicated
!
! Run RIP on the asynchronous lines because few implementations of SLIP
! understand IGRP. Run IGRP on the Ethernet (and in the local network).
!
router igrp 110
 network 10.11.12.0
! Send routes from the asynchronous lines on the production network.
 redistribute RIP
! Do not send IGRP updates on the asynchronous interfaces.
 passive-interface async 1
!
 router RIP
 network 10.11.12.0
 redistribute igrp
 passive-interface ethernet 0
! Consider filtering everything except a default route from the routing
! updates sent on the (slow) asynchronous lines.
 distribute-list 1 out
 ip unnumbered async 2
 async dynamic routing
```

IGRP Configuration Example

In the following example, only the Interior Gateway Routing Protocol (IGRP) TCP/IP routing protocol is running; it is assumed that the systems that are dialing in to use routing will either support IGRP or have some other method (for example, a static default route) of determining that the router is the best place to send most of its packets.

```
router igrp 111
 network 10.11.12.0
interface ethernet 0
 ip address 10.11.12.92 255.255.255.0
!
interface async 1
 async default ip address 10.11.12.96
 async dynamic routing
 ip tcp header-compression passive
 ip unnumbered ethernet 0

line 1
 modem ri-is-cd
```

Configuring Asynchronous Serial Traffic over UDP

This chapter describes how to communicate with a modem using the Asynchronous Serial Traffic over UDP feature in the following main sections:

- UDPTN Overview
- How to Configure Asynchronous Serial Traffic over UDP

See the "Configuration Examples for UDPTN" section for configuration examples.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

For a complete description of the UDP commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

UDPTN Overview

The Asynchronous Serial Traffic over UDP feature provides the ability to encapsulate asynchronous data into User Datagram Protocol (UDP) packets and then unreliably send this data without needing to establish a connection with a receiving device. This process is referred to as UDP Telnet (UDPTN), although it does not—and cannot—use the Telnet protocol. UDPTN is similar to Telnet in that both are used to send data, but UDPTN is unique in that it does not require that a connection be established with a receiving device. You load the data that you want to send through an asynchronous port, and then send it, optionally, as a multicast or a broadcast. The receiving device(s) can then receive the data whenever it wants. If the receiver ends reception, the transmission is unaffected.

The Asynchronous Serial Traffic over UDP feature provides a low-bandwidth, low-maintenance method to unreliably deliver data. This delivery is similar to a radio broadcast: It does not require that you establish a connection to a destination; rather, it sends the data to whatever device wants to receive it. The receivers are free to begin or end their reception without interrupting the transmission.

It is a low-bandwidth solution for delivering streaming information for which lost packets are not critical. Such applications include stock quotes, news wires, console monitoring, and multiuser chat features.

This feature is particularly useful for broadcast, multicast, and unstable point-to-point connections. This feature may not work as expected when there are multiple users on the same port number in a nonmulticast environment. The same port must be used for both receiving and sending.

How to Configure Asynchronous Serial Traffic over UDP

To configure the Asynchronous Serial Traffic over UDP feature, perform the tasks described in the following sections:

- Preparing to Configure Asynchronous Serial Traffic over UDP (Required)
- Configuring a Line for UDPTN (Required)
- Enabling UDPTN (Required)
- Verifying UDPTN Traffic (Optional but Recommended)

See the "Configuration Examples for UDPTN" section at the end of this chapter for multicast, broadcast, and point-to-point UDPTN configuration examples.

Preparing to Configure Asynchronous Serial Traffic over UDP

When configuring the Asynchronous Serial Traffic over UDP feature for multicast transmission, you must configure IP multicast routing for the entire network that will receive or propagate the multicasts. When configuring the feature for broadcast transmission, you must configure broadcast flooding on the routers between network segments. Refer to the "Configuring IP Multicast Routing" chapter of this guide for information on how to configure IP multicast routing. See the section "Configuring Broadcast Packet Handling" in the *Cisco IOS IP Configuration Guide* for information on how to configure broadcast flooding.

Configuring a Line for UDPTN

To configure the line that will be used to send or receive UDP packets, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line line-number`	Enters line configuration mode for the line number specified.
Step 2	`Router(config-line)# transport output udptn`	Enables the line to transport UDP packets.
Step 3	`Router(config-line)# dispatch-timeout 1000`	Sends packets every 1000 milliseconds.
Step 4	`Router(config-line)# dispatch-character 13`	Sends packets after every new line.
Step 5	`Router(config-line)# no session-timeout`	Disables timeout connection closing.

Enabling UDPTN

There are two methods of enabling UDPTN. You can manually enable UDPTN when you want to begin transmission or reception, or you can configure the router to automatically enable UDPTN when a connection is made to the line.

To manually enable UDPTN and begin UDPTN transmission or reception, use the following command in EXEC mode:

Command	Purpose
`Router# udptn ip-address [port] [/transmit] [/receive]`	Enables UDPTN to the specified IP address (optionally, using the specified port). Use the **/transmit** or **/receive** keyword if the router will only be sending or receiving UDPTN.

To automatically enable UDPTN when a connection is made to the line, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line line-number`	Enters line configuration mode for the line number specified.
Step 2	`Router(config-line)# autocommand udptn ip-address [port] [/transmit] [/receive]`	Enables UDPTN automatically when a connection is made to the line (optionally, using the specified port). Use the **/transmit** or **/receive** keyword if the router will only be sending or receiving UDPTN.

Verifying UDPTN Traffic

To verify that UDPTN is enabled correctly, perform the following steps:

Step 1 Enable UDPTN debugging by using the **debug udptn** EXEC command.

Step 2 Enable UDPTN by using the **udptn** *ip-address* EXEC command, and then observe the debug output.

The following debug output shows a UDPTN session being successfully established and then disconnected.

```
Router# debug udptn
Router# udptn 172.16.1.1
Trying 172.16.1.1 ... Open

*Mar  1 00:10:15.191:udptn0:adding multicast group.
*Mar  1 00:10:15.195:udptn0:open to 172.16.1.1:57 Loopback0jjaassdd
*Mar  1 00:10:18.083:udptn0:output packet w 1 bytes
*Mar  1 00:10:18.087:udptn0:Input packet w 1 bytes
Router# disconnect
Closing connection to 172.16.1.1 [confirm] y
Router#
```

Step 3 While the **udptn** command is enabled, enter the **show ip socket** command to verify that the socket being used for UDPTN opened correctly.

```
Router# show ip socket
Proto    Remote      Port          Local       Port  In  Out  Stat TTY OutputIF
 17    --listen--             172.21.14.90      67   0    0     89      0
 17      0.0.0.0      520     172.21.14.90     520   0    0      1      0
 17      1.1.1.2       57        1.1.1.1        57   0    0     48      0
 17     224.1.1.1      57        1.2.2.2        57   0    0     48      0 Loopback0
```

Configuration Examples for UDPTN

This section provides the following UDPTN configuration examples:

- Multicast UDPTN Example
- Broadcast UDPTN Example
- Point-to-Point UDPTN Example

Multicast UDPTN Example

These configurations are for multicast UDPTN. The router that is multicasting does not require a multicast configuration—it simply sends to the multicast IP address.

Router That Is Multicasting

```
ip multicast-routing
interface ethernet 0
 ip address 10.1.1.1 255.255.255.0
 ip pim dense-mode
!
line 5
 no session-timeout
 transport output udptn
 dispatch-timeout 10000
 dispatch-character 13
 modem in
 autocommand udptn 172.1.1.1 /transmit
```

Receiving Routers

```
ip multicast-routing
interface ethernet 0
 ip address 10.99.98.97 255.255.255.192
 ip pim dense-mode
!
line 0 16
 transport output udptn telnet lat rlogin
 autocommand udptn 172.1.1.1 /receive
```

Broadcast UDPTN Example

These configurations are for broadcast UDPTN. This is the simplest method to send to multiple receivers. The broadcasting router sends to the broadcast IP address, and any router that wants to receive the transmission simply connects to the broadcast IP address by using the **udptn** command.

Router That Is Broadcasting

```
interface ethernet 0
 ip address 10.1.1.1 255.255.255.0
!
line 5
 no session-timeout
 transport output udptn
 dispatch-timeout 10000
 dispatch-character 13
 modem in
 autocommand udptn 255.255.255.255 /transmit
```

Receiving Routers

```
interface ethernet 0
 ip address 10.99.98.97 255.255.255.192
!
line 0 16
 transport output udptn telnet lat rlogin
 autocommand udptn 255.255.255.255 /receive
```

Point-to-Point UDPTN Example

These configurations are for two routers in mobile, unstable environments that wish to establish a bidirectional asynchronous tunnel. Because there is no way to ensure that both routers will be up and running when one of the routers wants to establish a tunnel, they cannot use connection-dependent protocols like Telnet or local area transport (LAT). They instead use the following UDPTN configurations. Each router is configured to send to and receive from the IP address of the other. Because both routers will be sending and receiving, they do not use the **/transmit** or **/receive** keywords with the **udptn** command.

Router A

```
interface ethernet 0
 ip address 10.54.46.1 255.255.255.192
!
line 5
 no session-timeout
 transport output udptn
 dispatch-timeout 10000
 dispatch-character 13
 modem in
 autocommand udptn 10.54.46.2
```

Router B

```
interface ethernet 0
 ip address 10.54.46.2 255.255.255.192
!
line 10
 no session-timeout
 transport output udptn
 dispatch-timeout 10000
 dispatch-character 13
 modem in
 autocommand udptn 10.54.46.1
```

Modem Configuration and Management

Overview of Modem Interfaces

This chapter describes modem interfaces in the following sections:

- Cisco Modems and Cisco IOS Modem Features
- Cisco IOS Modem Components
- Logical Constructs in Modem Configurations

See the chapter "Overview of Dial Interfaces, Controllers, and Lines" for more information about Cisco asynchronous serial interfaces.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

For a complete description of the modem support commands in this chapter, refer to the *Cisco IOS Modem Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Cisco Modems and Cisco IOS Modem Features

Deciding which asynchronous features to use, to some degree, depends on your hardware configuration. All Cisco access servers must have their asynchronous interfaces and lines configured for network protocol support. Commands entered in asynchronous interface mode configure protocol-specific parameters for asynchronous interfaces, whereas commands entered in line configuration mode configure the physical and logical aspects for the same port.

Modems inside high-end access servers need a localized modem country code. This code is projected from the Cisco IOS software to the onboard modems using the **modem country** {**mica** | **microcom_hdms**} *country* command. The following are high-end access servers: Cisco AS5800, Cisco AccessPath, Cisco AS5300, and the Cisco AS5200.

Modems externally attached to low-end access servers need to receive initialization strings from the **modem autoconfigure discovery** command. For troubleshooting tips, see the section "External Modems on Low-End Access Servers" in the chapter "Configuring and Managing External Modems." The following are low-end access servers: Cisco AS2511-RJ, Cisco AS2509-RJ, Cisco 2509, Cisco 2511, and the Cisco 2512.

Figure 12 shows a Cisco AS2511-RJ access server. Figure 13 shows a Cisco AS5300 access server. Notice that modems are either inside or outside the chassis, depending on the product model.

Figure 12 Cisco AS2511-RJ Access Server

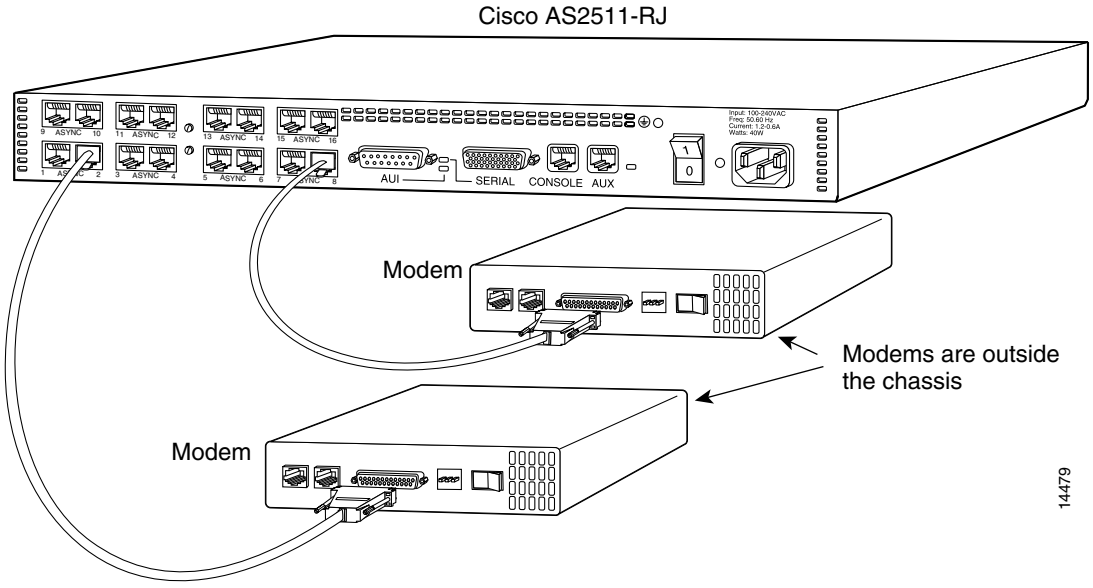
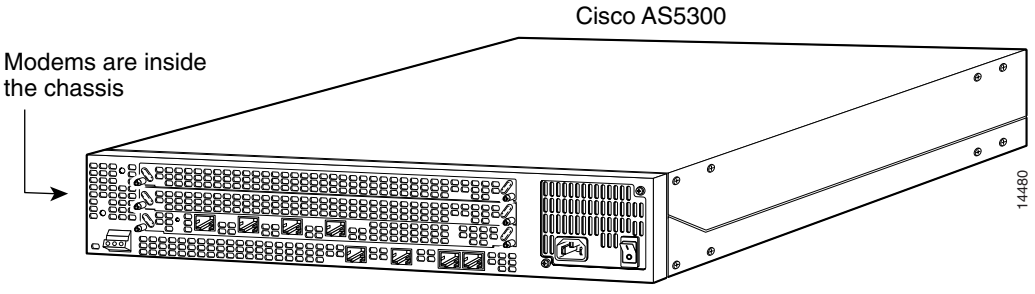


Figure 13 Cisco AS5300 Access Server

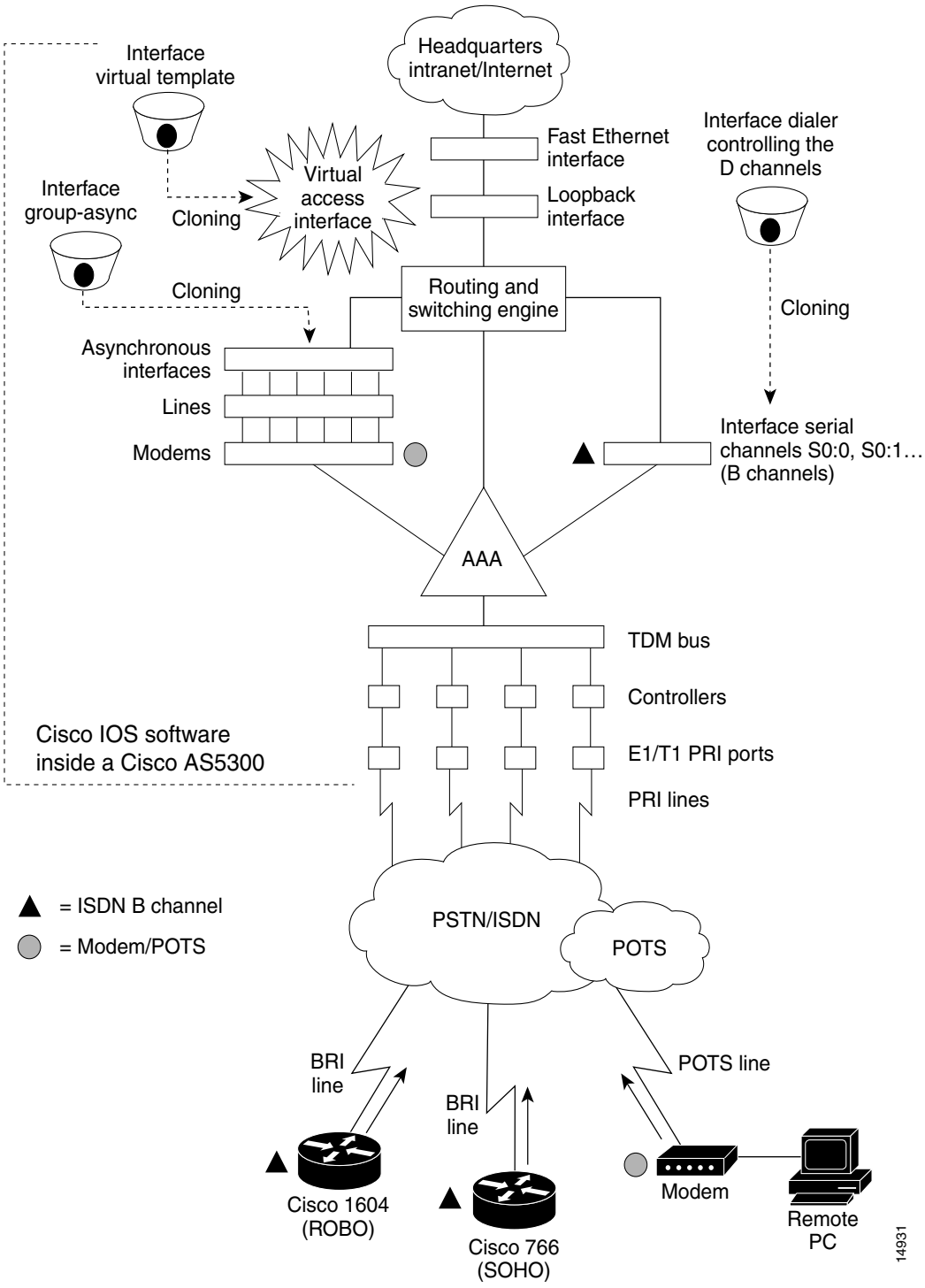


Cisco IOS Modem Components

Different components inside Cisco IOS software work together to enable remote clients to dial in and send packets. Figure 14 shows one Cisco AS5300 access server that is receiving calls from a remote office, branch office (ROBO); small office, home office (SOHO); and modem client.

Depending on your network scenario, you may encounter all of the components in Figure 14. For example, you might decide to create a virtual IP subnet by using a loopback interface. This step saves address space. Virtual subnets can exist inside devices that you advertise to your backbone. In turn, IP packets get relayed to remote PCs, which route back to the central site.

Figure 14 Cisco IOS Modem Concepts



Logical Constructs in Modem Configurations

A logical construct stores core protocol characteristics to assign to physical interfaces. No data packets are forwarded to a logical construct. Cisco uses three types of logical constructs in its access servers and routers. These constructs are described in the following sections:

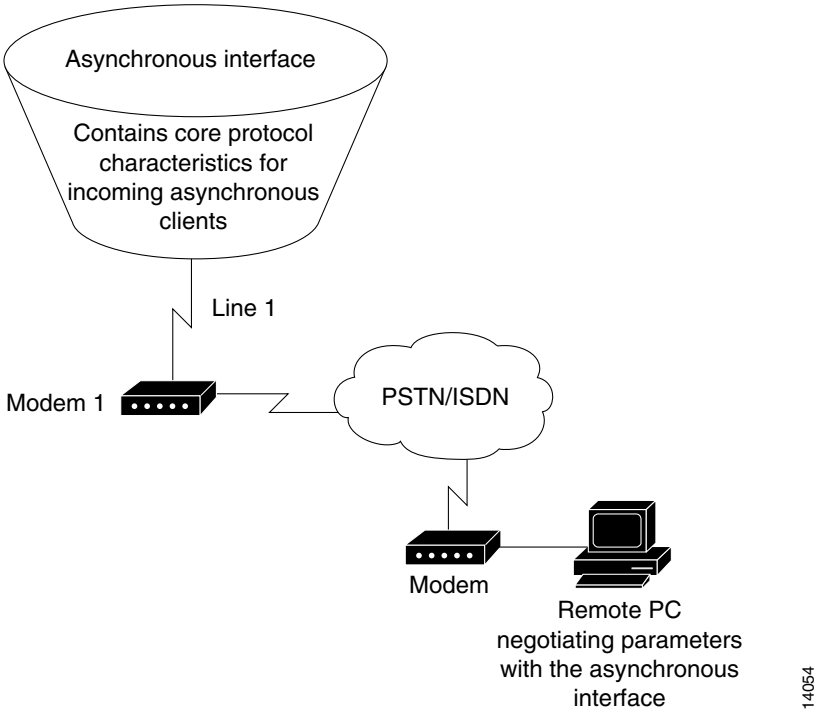
- Asynchronous Interfaces
- Group Asynchronous Interfaces
- Modem Lines and Asynchronous Interfaces

Asynchronous Interfaces

An asynchronous interface assigns network protocol characteristics to remote asynchronous clients that are dialing in through physical terminal lines and modems. (See Figure 15.)

Use the **interface async** command to create and configure an asynchronous interface.

Figure 15 Logical Construct for an Asynchronous Interface



To enable clients to dial in, you must configure two asynchronous components: asynchronous lines and asynchronous interfaces. Asynchronous interfaces correspond to physical terminal lines. For example, asynchronous interface 1 corresponds to tty line 1.

Commands entered in asynchronous interface mode configure protocol-specific parameters for asynchronous interfaces, whereas commands entered in line configuration mode configure the physical aspects for the same port.

Specifically, you configure asynchronous interfaces to support PPP connections. An asynchronous interface on an access server or router can be configured to support the following functions:

- Network protocol support such as IP, Internet Protocol Exchange (IPX), or AppleTalk
- Encapsulation support such as PPP
- IP client addressing options (default or dynamic)
- IPX network addressing options
- PPP authentication
- ISDN BRI and PRI configuration

For additional information about configuring asynchronous interfaces, see the "Overview of Dial Interfaces, Controllers, and Lines" chapter.

Group Asynchronous Interfaces

A group asynchronous interface is a parent interface that stores core protocol characteristics and projects them to a specified range of asynchronous interfaces. Asynchronous interfaces clone protocol information from group asynchronous interfaces. No data packets arrive in a group asynchronous interface.

By setting up a group asynchronous interface, you also eliminate the need to repeatedly configure identical configuration information across several asynchronous interfaces. For example, on a Cisco AS5300 one group asynchronous interface is used instead of 96 individual asynchronous interfaces. (See Figure 16.)

The following example shows a group asynchronous configuration for a Cisco AS5300 access server loaded with one 4-port ISDN PRI card and 96 MICA modems:

```
Router(config)# interface group-async 1
Router(config-if)# ip unnumbered loopback 0
Router(config-if)# encapsulation ppp
Router(config-if)# async mode interactive
Router(config-if)# peer default ip address pool dialin_pool
Router(config-if)# no cdp enable
Router(config-if)# ppp authentication chap pap dialin
Router(config-if)# group-range 1 96
```

To configure multiple asynchronous interfaces at the same time (with the same parameters), you can assign each asynchronous interface to a group and then configure the group. Configurations throughout this guide configure group asynchronous interfaces, rather than each interface separately.

If you want to configure different attributes on different asynchronous interfaces, do not assign them to the group or assign different interfaces to different groups. After assigning asynchronous interfaces to a group, you cannot configure these interfaces separately. For example, on a Cisco AS5300 access server in a T1 configuration, you could assign asynchronous interfaces 1 to 48 as part of one group (such as group-async1) and asynchronous interfaces 49 to 96 as part of another group (group-async2). You can also use the **member** command to perform a similar grouping function.

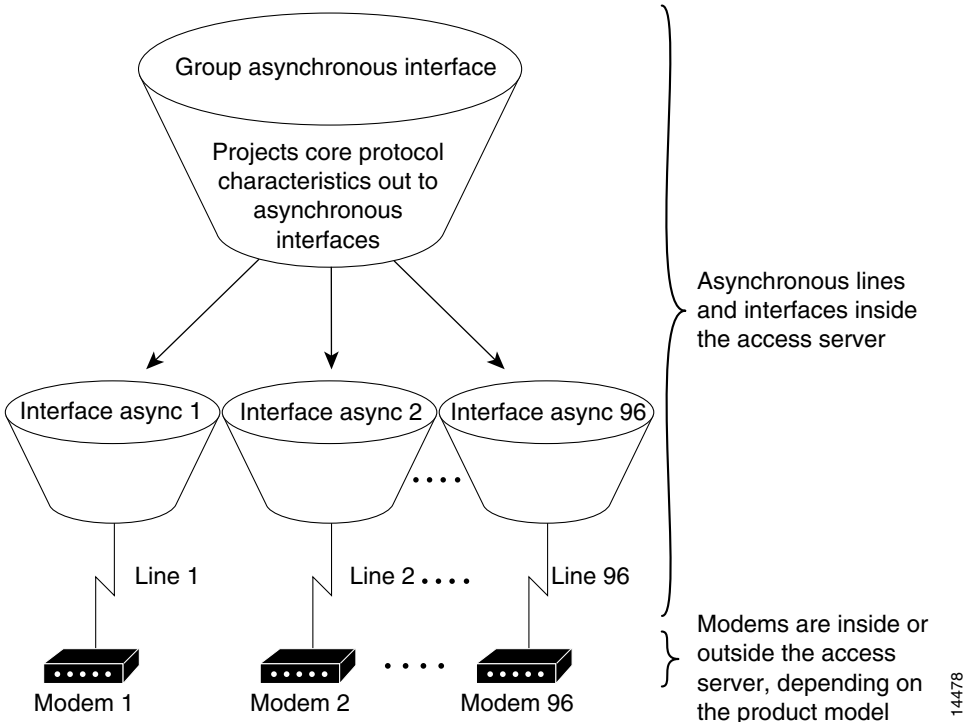
Modem Lines and Asynchronous Interfaces

Modems attach to asynchronous lines, which in turn attach to asynchronous interfaces. Depending on the type of access server you have, these components appear outside or inside the physical chassis. Figure 16 shows the logical relationships among modems, asynchronous lines, asynchronous interfaces, and group asynchronous interfaces. All these components work together to deliver packets as follows:

- Asynchronous calls come into the modems from the "plain old telephone service" (POTS) or Public Switched Telephone Network (PSTN).
- Modems pass packets up through asynchronous lines.
- Asynchronous interfaces clone their configuration information from group asynchronous interfaces.

Note The number of interfaces and modems varies among access server product models.

Figure 16 Modems, Lines, and Asynchronous Interfaces



Use the **interface group-async** command to create and configure a group asynchronous interface. The following example shows a group asynchronous configuration for a Cisco AS5300 access server loaded with one 4-port ISDN PRI card and 96 MICA modems:

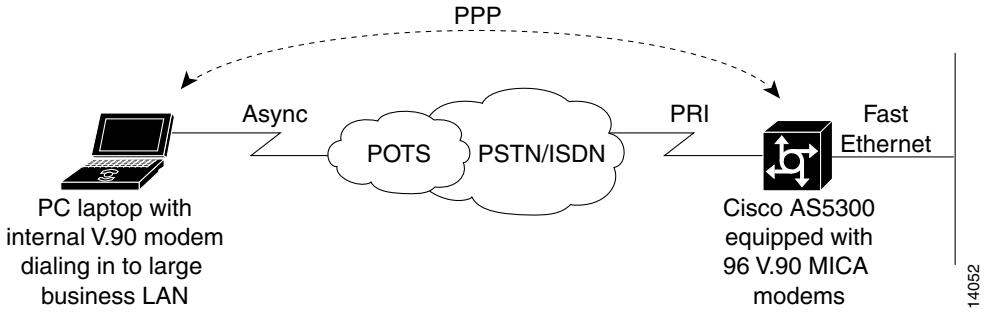
```
Router(config)# interface group-async 1
Router(config-if)# ip unnumbered loopback 0
Router(config-if)# encapsulation ppp
Router(config-if)# async mode interactive
Router(config-if)# peer default ip address pool dialin_pool
Router(config-if)# no cdp enable
Router(config-if)# ppp authentication chap pap dialin
Router(config-if)# group-range 1 96
```

Modem Calls

Modem calls travel through traditional telephone and ISDN lines. Regardless of the media used, these calls are initiated by a modem and terminate on another modem at the remote end.

Figure 17 shows a remote laptop using a V.90 internal modem to dial in to a Cisco AS5300 access server, which is loaded with 96 internal V.90 MICA technologies modems.

Figure 17 Remote Node Dialing In to a Cisco AS5300 Access Server



Asynchronous Line Configuration

Asynchronous line configuration commands configure ports for the following options:

- Physical layer options such as modem configuration
- Security for login in EXEC mode
- AppleTalk Remote Access (ARA) protocol configuration (PPP is configured in interface configuration mode)
- Autoselect to detect incoming protocols (ARA and PPP)

To enter line configuration mode, first connect to the console port of the access server and enter privileged EXEC mode. Then enter global configuration mode and finally enter line configuration mode for the asynchronous lines that you want to configure. The following example shows how you enter line configuration mode for lines 1 through 16:

```
Router> enable
Router# configure terminal
Router(config)# line 1 16
Router(config-line)#
```

Absolute Versus Relative Line Numbers

When you enter line configuration mode, you can specify an absolute line number or a relative line number. For example, absolute line number 20 is vty 2 (line 18 is vty 0). Referring to lines in a relative format is often easier than attempting to recall the absolute number of a line on a large system. Internally, the router uses absolute line numbers.

On all routers except the Cisco AS5350, AS5400, AS5800, AS5850 access servers, you can view all of the absolute and relative line numbers using the **show users all** EXEC command.

In the following sample display, absolute line numbers are listed at the far left. Relative line numbers are in the third column, after the line type. The second virtual terminal line, vty 1, is absolute line number 3. Compare the line numbers in this sample display to the output from the **show line** command.

```
Line       User     Host(s)              Idle Location
0 con 0
1 aux 0
2 vty 0             incoming             0 SERVER.COMPANY.COM
3 vty 1
4 vty 2
5 vty 3
6 vty 4
```

On the Cisco AS5350, AS5400, AS5800, AS5850 access servers, you can view the absolute and relative line numbers with the following commands:

- **show users all | exclude tty | interface** to show the non-internal modem lines
- **show controller async | include tty** to show the internal modem lines

The following example shows the information displayed with the **show users all | exclude tty|Interface** command:

```
Router# show users all | exclude tty | Interface
    Line       User       Host(s)              Idle       Location
*  0 con 0                 idle                 00:00:00
   1 aux 0                                      00:00:00
   2 vty 0                                      00:00:00
   3 vty 1                                      00:00:00
   4 vty 2                                      00:00:00
   5 vty 3                                      00:00:00
   6 vty 4                                      00:00:00
```

The following example shows the information displayed with the **show controller async | include tty** command:

Router# show controller async | include tty

```
Controller information for Async2/00 (tty324)
Controller information for Async2/01 (tty325)
Controller information for Async2/02 (tty326)
.
.
.
```

Compare the line numbers in this sample display to the output from the **show line** command.

Line and Modem Numbering Issues

The tty line numbering scheme used by your access server or router is specific to your product and its hardware configuration. Refer to the product-specific documentation that came with your product for line numbering scheme information.

For example, the Cisco AS5200 access server has tty lines that map directly to integrated modems, as shown in Table 5. Depending on the shelf, slot, and port physical architecture of the access server, the modem and tty line number schemes will change.

As shown in Table 5, physical terminal lines 1 through 24 directly connect to modems 1/0 through 1/23, which are installed in the first chassis slot in this example. Physical terminal lines 25 through 48 directly connect to modems 2/0 through 2/23, which are installed in the second slot.

Table 5 tty Lines Associated with Cisco AS5200 Modems

tty Line	Slot/Modem Number	tty Line	Slot/Modem Number
1	1/0	25	2/0
2	1/1	26	2/1
3	1/2	27	2/2
4	1/3	28	2/3
5	1/4	29	2/4
6	1/5	30	2/5
7	1/6	31	2/6
8	1/7	32	2/7
9	1/8	33	2/8
10	1/9	34	2/9
11	1/10	35	2/10
12	1/11	36	2/11
13	1/12	37	2/12
14	1/13	38	2/13
15	1/14	39	2/14
16	1/15	40	2/15
17	1/16	41	2/16
18	1/17	42	2/17
19	1/18	43	2/18
20	1/19	44	2/19
21	1/20	45	2/20
22	1/21	46	2/21
23	1/22	47	2/22
24	1/23	48	2/23

Decimal TCP Port Numbers for Line Connections

Connections to an individual line are most useful when a dial-out modem, parallel printer, or serial printer is attached to that line. To connect to an individual line, the remote host or terminal must specify a particular TCP port on the router.

If reverse XRemote is required, the port is 9000 (decimal) plus the decimal value of the line number.

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

If Telnet protocols are required, the port is 2000 (decimal) plus the decimal value of the line number. The Telnet protocol might require that Return characters be translated into Return and line-feed character pairs. You can turn off this translation by specifying the Telnet binary mode option. To specify this option, connect to port 6000 (decimal) plus the decimal line number.

For example, a laser printer is attached to line 10 of a Cisco 2511 router. Such a printer usually uses XON/XOFF software flow control. Because the Cisco IOS software cannot receive an incoming connection if the line already has a process, you must ensure that an EXEC session is not accidentally started. You must, therefore, configure it as follows:

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

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

Signal and Flow Control Overview

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

Dialup modems that operate over normal telephone lines at speeds of 28800 bps use hardware flow control to stop the data from reaching the host by toggling an EIA/TIA-232 signal when their limit is reached.

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

The Cisco IOS software supports hardware flow control on its CTS input signal, which is also used by the normal modem handshake.

Configuring and Managing Integrated Modems

The Cisco IOS software provides commands that manage modems that reside inside access servers or routers in the form of modem cards. This chapter describes the modem management tasks. It includes the following main sections:

- Modems and Modem Feature Support
- Managing Modems
- Configuration Examples for Modem Management

For additional instructions for configuring Cisco access servers, see the chapter "Configuring and Managing Cisco Access Servers and Dial Shelves" in this publication.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Modem initialization strings are listed in the "Modem Initialization Strings" appendix. For a complete description of the commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Modems and Modem Feature Support

The Cisco IOS software supports three types of integrated modems for Cisco access servers and access routers:

- Modem ISDN channel aggregation (MICA) digital modem
- NextPort digital modem
- NM-AM network module analog modem

Table 6 lists device support for each of the Cisco access server hardware platforms.

Table 6 Cisco IOS Modems and Modem Feature Support

Device Support	Cisco AS5300	Cisco AS5350	Cisco AS5400	Cisco AS5800	Cisco 2600/3600 Series Routers
Integrated modems	6- and 12-port MICA	60-port NextPort CSM v6DFC	108-port NextPort CSM v6DFC	72- and 144-port MICA 324-port NextPort CSM v6DFC	6-port, 12-port, 18-port, 24-port, or 30-port MICA NM-DM 8- and 16-port analog NM-AM
V.90	Yes	Yes	Yes	Yes	Yes with NM-DM
V.110	Yes	Yes	Yes	Yes	Yes with NM-DM
V.120	No, CPU only	Yes	Yes	Yes with 324-port NextPort[1] CSM v6DFC	No, CPU only

1. For more detailed information regarding the V.120 functionalities that are supported both by NextPort and Cisco IOS software, see the section "V.120 Bit Rate Adaptation Standard."

Note If the platform is using MICA technologies modems, the V.120 rate adaptation is done by CPU on vty lines like protocol translation sessions.

The following sections summarize the standards supported by modems in the Cisco access servers. See Table 7 through Table 10 for a summary and comparison of the Cisco IOS commands used for the MICA and NextPort modems.

V.90 Modem Standard

Study Group 16 of the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) developed the V.90 modem standard for multimedia systems. The V.90 standard describes a digital modem and analog modem pair for use on the public switched telephone network (PSTN). V.90 modems are designed for connections that are digital at one end and have only one digital-to-analog conversion. The V.90 standard is expected to be widely used for applications such as Internet and online service access. Download speeds of up to 56,000 bits per second (bps) are possible, depending on telephone line conditions, with upload speeds of up to 33,600 bps.

V.110 Bit Rate Adaption Standard

V.110 is a bit rate adaptation standard defined by the ITU that provides a standard method of encapsulating data over global system for mobile telecommunication (GSM) and ISDN networks. V.110 allows for reliable transport of asynchronous or synchronous data. V.110 adapts a low-speed connection

to an ISDN B channel allowing the remote station or terminal adapter to use the fast call setup times offered by ISDN. This feature allows V.110 calls to be originated and terminated over ISDN. It also enables GSM wireless connectivity.

V.110, as an alternative to V.120, provides DTE with V-series type interfaces with access to ISDN network by bit stuffing. Many V.110 devices are used in Europe and Japan. In Japan, MICA supports the Personal-Handyphone-System Internet Access Forum Standard (PIAFS) protocol, which is similar to V.110.

The V.110 implementation for calls on MICA modems is managed by special boardware and modem code, along with the appropriate Cisco IOS image, in a manner similar to other modulation standards. This MICA V.110 implementation provides V.110 user rates ranging from 600 bps to 38,400 bps.

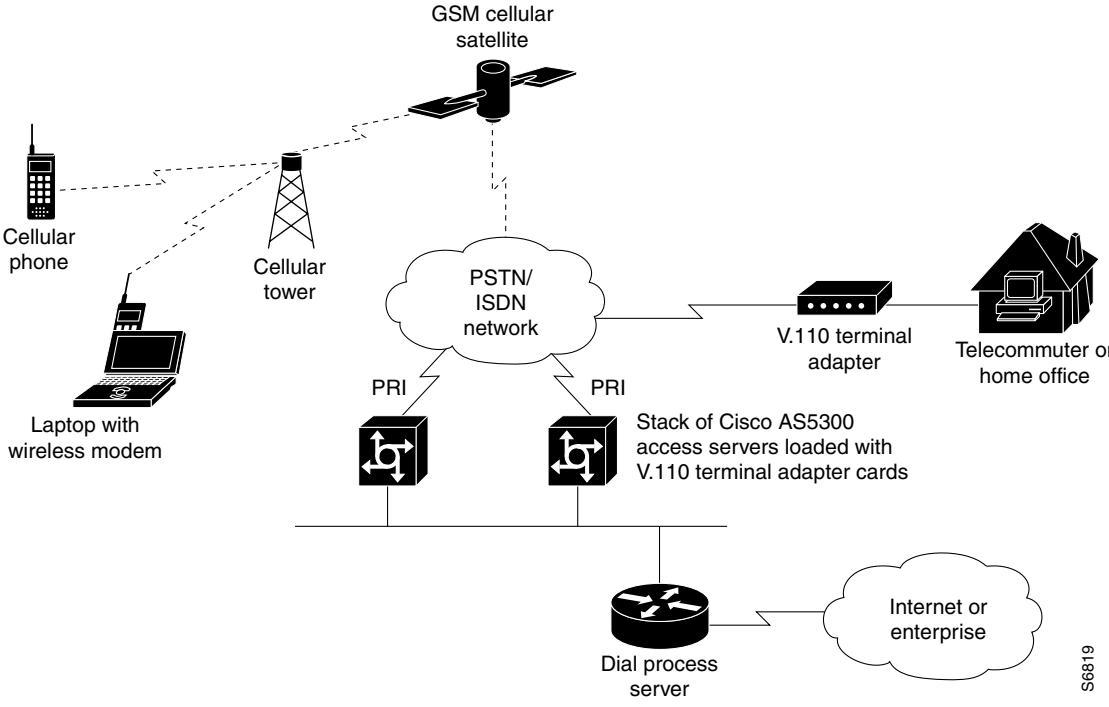
V.110 is supported on the following Cisco devices and network modules:

- Cisco AS5300-series access servers
- Cisco 3620, 3640, and 3660 access routers
- NM-6DM, NM-12DM, NM-18DM, NM-24DM, and NM-30DM network modules

The digital signal processors (DSPs) on the board can function as either modems or V.110 terminal adapters (or V.120 terminal adapters for NextPort DSPs). Based on the ISDN Q.931 bearer capability information element, the Cisco IOS software configures the DSP to treat the incoming call as a modem call, a V.110 call, or a V.120 call.

Figure 18 shows a dial-in scenario for how V.110 technology can be used with a stack of Cisco AS5300-series access servers.

Figure 18 V.110 Dial-In Scenario Using a Stack of Cisco AS5300-Series Access Servers



V.120 Bit Rate Adaptation Standard

ITU-T Recommendation V.120 revised by the ITU-T Study Group 14. V.120 describes a standard that can be used for adapting terminals with non-ISDN standard network interfaces to an ISDN. It is intended to be used between two terminal adapter (TA) functional groups, between two ISDN terminal (TE1) functional groups, between a TA and a TE1, or between either a TA or TE1 and an interworking facility inside a public or private ISDN.

V.120 allows for reliable transport of synchronous, asynchronous, or bit transparent data over ISDN bearer channels. Cisco provides three V.120 support features for terminal adapters that do not send the low-layer compatibility fields or bearer capability V.120 information:

- Answer all incoming calls as V.120—Static configuration used when all remote users have asynchronous terminals and need to connect with a vty on the router.
- Automatically detect V.120 encapsulation—Encapsulation dynamically detected and set.
- Enable V.120 support for asynchronous access over ISDN.

For terminal adapters that send the low-layer compatibility or bearer capability V.120 information, mixed V.120 and ISDN calls are supported. No special configuration is required.

V.120 is a digital rate adaptation and cannot be done on NM-AM network module analog modems. MICA DSP firmware does not have the code to terminate V.120 calls.

NextPort supports only a subset of V.120 functionalities that are supported by Cisco IOS software. Therefore, certain V.120 calls still will need to be terminated on the CPU, even if the chassis has available NextPort modems.

Managing Modems

To manage modems, perform the tasks in the following sections; the tasks you need to perform depend upon the type and needs of your system:

- Managing SPE Firmware
- Configuring Modems in Cisco Access Servers
- Configuring Cisco Integrated Modems Using Modem Attention Commands
- Configuring Modem Pooling
- Configuring Physical Partitioning
- Configuring Virtual Partitioning
- Configuring Call Tracker
- Configuring Polling of Link Statistics on MICA Modems
- Configuring MICA In-Band Framing Mode Control Messages
- Enabling Modem Polling
- Setting Modem Poll Intervals
- Setting Modem Poll Retry
- Collecting Modem Statistics
- Troubleshooting Using a Back-to-Back Modem Test Procedure
- Clearing a Direct Connect Session on a Microcom Modem

- Displaying Local Disconnect Reasons
- Removing Inoperable Modems
- Busying Out a Modem Card
- Monitoring Resources on Cisco High-End Access Servers

Managing SPE Firmware

You can upgrade your modem firmware to the latest NextPort Service Processing Element (SPE) firmware image available from Cisco. The SPE firmware image is usually retrieved from Cisco.com. You must first copy the SPE image from a TFTP server to flash memory using the **copy tftp flash** command. You then configure the firmware upgrade using the **firmware location** and **firmware upgrade** SPE configuration commands. The **firmware location** command specifies the location of the firmware file and downloads the firmware to an SPE or a range of SPEs, according to the schedule you selected for the firmware upgrade method using the **firmware upgrade** command.

The modem firmware upgrade commands must be saved into the system configuration using the **write memory** command; otherwise, at the next reboot downloading of the specified firmware will not occur.

To upgrade SPE firmware, use the following commands:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.

Step 2	AS5400:
	`Router(config)# spe slot/spe`
	or
	`Router(config)# spe slot/spe slot/spe`
	
	AS5800:
	`Router(config)# spe shelf/slot/spe`
	or
	`Router(config)# spe shelf/slot/spe shelf/slot/spe`
		Enters SPE configuration mode. You can choose to configure a range of SPEs by specifying the first and last SPE in the range.

Step 3	`Router(config-spe)# firmware upgrade {busyout | download-maintenance | reboot}`	Specifies the upgrade method. Three methods of upgrade are available. The **busyout** keyword waits until all calls are terminated on an SPE before upgrading the SPE to the designated firmware. The **download-maintenance** keyword upgrades the firmware during the download maintenance time. The **reboot** keyword requests the access server to upgrade firmware at the next reboot.

	Command	Purpose
Step 4	`Router(config-spe)# firmware location [IFS:[/]]filename`	Specifies the SPE firmware file in flash memory to use for the selected SPEs. Allows you to upgrade firmware for SPEs after the new SPE firmware image is copied to your flash memory. The Cisco IOS file specification (IFS) can be any valid IFS on any local file system. Use the **dir all-filesystems** EXEC command to display legal IFSs. Examples of legal IFS specifications include: • **bootflash:**—Loads the firmware from a separate flash memory device. • **flash:**—Loads the firmware from the flash NVRAM located within the router. • **system:/**—Loads the firmware from a built-in file within the Cisco IOS image. The optional forward slash (/) and system path must be entered with this specification. • *filename*—The name of the desired firmware file (for example, mica-modem-pw.2.7.3.0.bin). If the **system** keyword is specified, enter the path to the filename you want to download.
Step 5	`Router(config-spe)# exit`	Exits SPE configuration mode.
Step 6	`Router(config)# exit`	Exits global configuration mode.
Step 7	`Router# copy running-config startup-config`	Saves your changes.

Note As soon as a firmware file is specified, the downloading begins. Do not specify all modems and then go into an upgrade process on a busy router. The modems that are not busy will all be marked busy and the server will wait until all the modems on each of the given cards are free before upgrading the multiple-port cards. The only way to clear this situation is to start disconnecting users with a **clear** command. Normally, groups of modems are specified in scripts with the spe slot/spe_begin and slot/spe_end statements, and upgrades are done in a rolling fashion.

Use the **show modem version** and **show spe version** commands to verify that the modems are running the portware version you specified.

The following example shows how to enter the SPE configuration mode, set the range of SPEs, specify the firmware file location in flash memory, download the file to the SPEs, and display a status report using the **show spe** EXEC command:

```
Router# configure terminal
Router(config)# spe 7/0 7/17
Router(config-spe)# firmware upgrade busyout
Router(config-spe)# firmware location flash:np_6_75
Started downloading firmware flash:np_6_75.spe
Router(config-spe)# exit
Router(config)# exit
Router# show spe 7
.
.
.
```

```
                    SPE          SPE     SPE  SPE   Port         Call
SPE#    Port #      State        Busyout Shut Crash State        Type
7/00    0000-0005   ACTIVE             1    0     0 BBBBBB       ______
7/01    0006-0011   DOWNLOAD           1    0     0 bbbbbb       ______
7/02    0012-0017   DOWNLOAD           1    0     0 bbbbbb       ______
7/03    0018-0023   DOWNLOAD           1    0     0 bbbbbb       ______
.
.
.
```

For information about upgrading Cisco 3600 Series and Cisco 3700 modems, see the *Cisco 3600 Series and Cisco 3700 Series Modem Portware Upgrade Configuration Note* at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3600/sw_conf/portware/5257d56k.htm .

Configuring Modems in Cisco Access Servers

To configure modem support for access servers such as the Cisco AS5300 and AS5800, perform the following tasks. The list describes which tasks are required and which are optional but recommended.

- Configuring Modem Lines (Required)
- Verifying the Dial-In Connection (Optional but Recommended)
- Troubleshooting the Dial-In Connection (Optional but Recommended)
- Configuring the Modem Using a Modemcap (Required)
- Configuring the Modem Circuit Interface (Required for Digital Modems)

Note See the chapter "Configuring and Managing Cisco Access Servers and Dial Shelves" for additional information about configuring Cisco AS5x00 series access servers.

Configuring Modem Lines

You must configure the modem lines and set the country code to enable asynchronous connections into your access server. To configure the modems and line, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	**MICA modems**
	`Router(config)# modem country mica country`
	NextPort SPE modems
	`Router(config)# spe country country`
	Microcom modems
	`Router(config)# modem country microcom_hdms country`
		Depending on the type of modems loaded in your access server, specifies the modem vendor and country code.[1] This step is only for the MICA, NextPort SPE, and Microcom modems in the Cisco AS5000 series access servers.
		Table 7 through Table 10 provide a summary and comparison of the Cisco IOS commands used for the MICA and NextPort modems.

Step 2	`Router(config)# line beginning-line-number ending-line-number`	Enters the number of modem lines to configure. Usually this range is equal to the number of modems in the access server. Use the **show line** EXEC command to see which lines are available.

	Command	Purpose
Step 3	`Router(config-line)# transport {input | output} {all | none}`	Specifies that connection protocols can be used when connecting to the line. For outgoing calls, choose the **output** option. For incoming calls, choose the **input** option. If you do not intend to dial out, choose the **none** option.
Step 4	`Router(config-line)# autoselect {arap | ppp | slip}`	Configures the line to automatically startup an AppleTalk Remote Access (ARA), PPP, and Serial Line Internet Protocol (SLIP) session. You can configure more than one protocol by entering multiple **autoselect** commands with the appropriate keyword.
Step 5	`Router(config-line)# autoselect during-login`	Configures the lines to display the username and password prompt as soon as the line is connected, rather than waiting until the user presses the Enter or Return key at the terminal.
Step 6	`Router(config-line)# login authentication dialin` or `Router(config-line)# login` *login-name* `Router(config-line)# password` *password*	Enables authentication across all asynchronous modem logins. Use the **login authentication dialin** command when authentication, authorization, and accounting (AAA) authentication has been enabled. Use the **login** and **password** commands to configure non-AAA user authentication.
Step 7	`Router(config-line)# modem dialin`	Configures the modem for only incoming calls.
Step 8	`Router(config-line)# exit`	Returns to global configuration mode.

1. For a comprehensive list of modem country codes, see the **modem country mica** command and the **modem country microcom_hdms** command in the *Cisco IOS Dial Technologies Command Reference*.

Verifying the Dial-In Connection

Before configuring any additional protocols for the line such as SLIP, PPP, or ARA, test whether the dial-in connection for the access server and modem are configured correctly for dial-in access,

Note The same configuration issues exist between the client DTE and client modem. Make sure that you have the correct EIA/TIA-232 cabling and modem initialization string for your client modem.

The following is an example of a successful connection from a PC using a known good modem to dial in to a Cisco access server:

```
at
OK
atdt9,5550101
CONNECT 14400/ARQ/V32/LAPM/V42BIS
User Access Verification
Username: user1
Password:
Router>
```

Troubleshooting the Dial-In Connection

Depending upon the problems you experience, take the appropriate action:

- If you are having problems making or receiving calls, make sure that you turned on the protocols for connecting to the lines and configured for incoming and outgoing calls.
- If the calls are not coming up at all, turn on modem debugging. Use the the modem debugging commands as follows:
  - The **debug modem** command enables debugging on the modem line.
  - The **debug modem csm** (or **debug csm modem)** command enables debugging for lines configured for digital modems.
  - The **debug isdn q931** command enables debugging for lines configured for the ISDN and Signaling System 7 (SS7) Q.931 protocols.
  - The **debug cas** command enables debugging for lines configured for channel-associated signaling (CAS).

  Following is a sample of how to enable and then disable Cisco IOS modem debugging commands on a network access server:

```
Router# debug modem
Router# debug modem csm
Router# debug isdn q931
Router# no debug modem
Router# no debug modem csm
Router# no debug isdn q931
```

- Enter the **debug modem ?** command for a list of additional modem debugging commands:

```
Router# debug modem ?
  b2b          Modem Special B2B
  csm          CSM activity
  maintenance  Modem maintenance activity
  mica         MICA Async driver debugging
  oob          Modem out of band activity
  tdm          B2B Modem/PRI TDM
  trace        Call Trace Upload
```

- Turn off the messages by entering the **no debug modem** command.

For more detailed information refer to the TAC Tech Notes document, *Troubleshooting Modems*, at the following URL: http://www.cisco.com/warp/public/471/index_14280.html

Configuring the Modem Using a Modemcap

Modems are controlled by a series of parameter settings (up to a limit of 128 characters) that are sent to the modem to configure it to interact with a Cisco device in a specified way. The parameter settings are stored in a database called a *modem capability* (modemcap). The Cisco IOS software contains defined modemcaps that have been found to properly initialize internal modems. Following are the names of some modemcaps available in the Cisco IOS software:

- cisco_v110—Cisco (NEC) internal V.110 TA (AS5200)
- mica—Cisco MICA HMM/DMM internal digital modem
- nextport—Cisco NextPort CSMV/6 internal digital modem
- microcom_hdms—Microcom HDMS chassis

- microcom_mimic—Cisco (Microcom) internal analog modem (NM-AM–2600/3600)
- microcom_server—Cisco (Microcom) V.34/56K internal digital modem (AS5200)

Enter these modemcap names with the **modem autoconfigure type** command.

For more information on creating and using modemcaps refer to the TAC Tech Notes documentation, *Recommended Modemcaps for Internal Digital and Analog Modems on Cisco Access Servers*, at the following URL: http://www.cisco.com/warp/public/471/recc_modemcaps.html

If your modem is not on this list and if you know what modem initialization string you need to use with it, you can create your own modemcap; see the following procedure, "Using the Modem Autoconfigure Type Modemcap Feature." To have the Cisco IOS determine what type of modem you have, use the **modem autoconfigure discovery** command to configure it, as described in the procedure "Using the Modem Autoconfigure Discovery Feature."

Note When configuring an internal modem, avoid using the Modem Autoconfigure Discovery feature because the feature can misdetect the internal modem type and cause the modem to start working in an unpredictable and unreproducable manner.

Using the Modem Autoconfigure Type Modemcap Feature

If you know what modem initialization string you need to use with your modem, you can create your own modemcap by performing the following steps.

Step 1 Use the **modemcap edit** command to define your own modemcap entry.

The following example defines modemcap MODEMCAPNAME:

```
Router(config)# modemcap edit MODEMCAPNAME miscellaneous &FS0=1&D3
```

Step 2 Apply the modemcap to the modem lines as shown in the following example:

```
Router# terminal monitor
Router# debug confmodem
Modem Configuration Database debugging is on
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#line 33 34
Router(config-line)#modem autoconfigure type MODEMCAPNAME
Jan 16 18:12:59.643: TTY34: detection speed (115200) response ---OK---
Jan 16 18:12:59.643: TTY34: Modem command:  --AT&FS0=1&D3--
Jan 16 18:12:59.659: TTY33: detection speed (115200) response ---OK---
Jan 16 18:12:59.659: TTY33: Modem command:  --AT&FS0=1&D3--
Jan 16 18:13:00.227: TTY34: Modem configuration succeeded
Jan 16 18:13:00.227: TTY34: Detected modem speed 115200
Jan 16 18:13:00.227: TTY34: Done with modem configuration
Jan 16 18:13:00.259: TTY33: Modem configuration succeeded
Jan 16 18:13:00.259: TTY33: Detected modem speed 115200
Jan 16 18:13:00.259: TTY33: Done with modem configuration
```

Note The report that is generated by the **debug confmodem** command can be misleading for the MICA and NextPort internal modems because these modems do not have Universal Asynchronous Receiver/Transmitter (UART) and exchange data with the CPU at speeds of hundreds of kbps.

Using the Modem Autoconfigure Discovery Feature

If you prefer that the modem software use its autoconfigure mechanism to configure the modem, use the **modem autoconfigure discovery** command.

The following example shows how to configure modem autoconfigure discovery mode:

```
Router# terminal monitor
Router# debug confmodem
Modem Configuration Database debugging is on
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# line 33 34
Router(config-line)# modem autoconfigure discovery
Jan 16 18:16:17.724: TTY33: detection speed (115200) response ---OK---
Jan 16 18:16:17.724: TTY33: Modem type is default
Jan 16 18:16:17.724: TTY33: Modem command:  --AT&F&C1&D2S0=1H0--
Jan 16 18:16:17.728: TTY34: detection speed (115200) response ---OK---
Jan 16 18:16:17.728: TTY34: Modem type is default
Jan 16 18:16:17.728: TTY34: Modem command:  --AT&F&C1&D2S0=1H0--
Jan 16 18:16:18.324: TTY33: Modem configuration succeeded
Jan 16 18:16:18.324: TTY33: Detected modem speed 115200
Jan 16 18:16:18.324: TTY33: Done with modem configuration
Jan 16 18:16:18.324: TTY34: Modem configuration succeeded
Jan 16 18:16:18.324: TTY34: Detected modem speed 115200
Jan 16 18:16:18.324: TTY34: Done with modem configuration
```

Configuring the Modem Circuit Interface

The next task to complete before using the integrated modem is to configure the modem circuit interface. The basic steps are outlined next:

- If the integrated modem is an analog modem, no further configuration is required; modem characteristics are set on the line.
- If the integrated modem is a digital modem, you can configure either the ISDN or CAS, as appropriate.
  - For ISDN BRI and PRI, you need to select the switch type and whether ISDN accepts incoming voice or data calls. If you configure a PRI, you will need to configure the T1 or E1 controller. See the chapter "Configuring ISDN BRI" in the "ISDN Configuration" part of this guide, and the chapter "Configuring ISDN PRI" in the "Signaling Configuration" part of this guide.
  - Configuring CAS is described in the chapter "Configuring ISDN PRI" in the Signaling Configuration part of this guide.

If you want to configure SS7, refer to Appendix G, "Configuring the Cisco SS7/C7 Dial Access Solution System," in the *Cisco IOS Voice, Video, and Fax Configuration Guide*.

Comparison of NextPort SPE and MICA Modem Commands

Table 7 through Table 10 compare the MICA and SPE commands.

Table 7 EXEC Commands: NextPort to MICA Command Comparison

NextPort SPE Commands	Purpose	MICA Modem Commands
clear port	Clears specified ports.	**clear modem**
clear port log	Clears all log entries for specified ports.	**clear modem log**

Table 7 EXEC Commands: NextPort to MICA Command Comparison (continued)

NextPort SPE Commands	Purpose	MICA Modem Commands
clear spe	Reboots all specified SPEs. All calls will be torn down.	none
clear spe counters	Clears all statistics.	**clear modem counters**
clear spe log	Clears all log entries for specified SPEs.	**clear modem log**
show port config	Displays configuration parameters for the current active session.	**show modem config**
show port modem calltracker	Displays port-level information for an active modem.	**show modem calltracker**
show port modem log	Displays the events generated by the modem sessions.	**show modem log**
show port modem test	Displays port modem test results.	**show modem test**
show port operational-status	Displays statistics for the current active session.	**show modem operational-status**
show spe	Displays the SPE status.	—
show spe log	Displays the SPE system log.	—
show spe modem active	Displays the statistics of all active calls on specified SPEs.	**show modem**
show spe modem csr	Displays the call success rate (CSR) for the specified SPE.	**show modem**
show spe modem disconnect-reason	Displays all modem disconnect reasons for the specified SPEs.	**show modem call-stats**
show spe modem high speed	Displays the total number of connections negotiated within each modulation or coder-decoder (codec) for a specific range of SPEs.	**show modem speed**
show spe modem high standard	Displays the total number of connections negotiated within each high modulation or codec for a specific range of SPEs or for all the SPEs.	—
show spe modem low speed	Displays the connect-speeds negotiated within each low-speed modulation or codec for a specific range of SPEs or for all the SPEs.	**show modem speed**
show spe modem low standard	Displays the total number of connections negotiated within each low modulation or codec for a specific range of SPEs or for all the SPEs.	—
show spe modem summary	Displays the modem service history statistics for specific SPEs.	**show modem**
show spe version	Displays all MICA and NextPort firmware versions stored in flash memory and the firmware assigned to each SPE.	**show modem mapping**

Table 8 SPE Configuration Commands: NextPort to MICA Command Comparison

NextPort SPE Commands	Purpose	MICA Modem Commands
busyout	Busies out active calls.	**modem busyout**
firmware location *filename*	Specifies the firmware file to be upgraded.	Already implemented on the Cisco AS5300 and Cisco AS5800 platforms.
firmware upgrade	Specifies the upgrade method.	Already implemented on the Cisco AS5300 platform.
port modem autotest[1]	Enables modem autotest.	**modem autotest**
shutdown	Tears down all active calls on the specified SPEs.	**modem shutdown**
spe	Configures the SPE.	Already implemented on the Cisco AS5300 and Cisco AS5800 platforms.
spe call-record	Generates a modem call record at the end of each call.	**modem call-record**
spe country	Sets the system country code.	**modem country**
spe log-size	Sets the maximum log entries for each port.	**modem buffer-size**
spe poll	Sets the statistic polling interval.	**modem poll**

1. Cisco does not recommend the use of the **modem autotest** or **port modem autotest** command. These commands may produce unexpected results including modems being marked out of service and unscheduled reloads. These commands have been removed in Cisco IOS Release 12.3.

Table 9 Port Configuration Commands: NextPort to MICA Command Comparison

NextPort SPE Commands	Purpose	MICA Modem Commands
busyout	Busies out a port.	**modem busyout**
default	Compares the value of the command to its default value.	**default modem**
port	Configures the port range.	**modem range**
shutdown	Shuts down a port.	**modem shutdown**

Table 10 Global Configuration Commands: NextPort to MICA Command Comparison

NextPort SPE CLI Commands	Purpose	MICA Modem CLI Commands
ds0 busyout-threshold	Defines a threshold to maintain a balance between the number of digital signal level 0s (DS0s) and modems.	**modem busyout-threshold**

Configuring Cisco Integrated Modems Using Modem Attention Commands

This section provides information about using modem attention (AT) command sets to modify modem configuration. It contains the following sections:

- Using Modem Dial Modifiers on Cisco MICA Modems (As required)
- Changing Configurations Manually in Integrated Microcom Modems (As required)
- Configuring Leased-Line Support for Analog Modems (As required)

Using Modem Dial Modifiers on Cisco MICA Modems

Dial modifiers permit multistage dialing for outbound modem calling through public and private switched telephone networks (PSTNs).

Note For additional information about dial modifiers for the MICA modems, search Cisco.com for the publication *AT Command Set and Register Summary for MICA Six-Port Modules*.

The Cisco NAS Modem Health feature is enabled by arguments to the **ATD AT** command. The **AT** prefix informs the network access server modem that commands are being sent to it, and the **D** (dial string or dial) suffix dials a telephone number, establishing a connection. With NAS Modem Health feature, you can enter the dial modifiers listed in Table 11 after the **D** in your dial string: **X**, **W**, and the comma (**,**) character. These modifiers had been previously accepted without error but ignored in Cisco MICA modems on Cisco AS5300 and Cisco AS5800 universal access servers.

Table 11 Dial Modifiers for Cisco MICA Modems

Dial Modifier	Definition
X	Switches to in-band dual tone multifrequency (DTMF) mode for any subsequent digits remaining in the **ATD** string. The **X** dial modifier has been added to serve as a delimiter for the host when the dial string is processed. It allows Cisco MICA portware to be used in many environments that do not support DTMF dialing (for example, PRI).
W	Waits for dial tone and then switches to in-band DTMF mode for any subsequent digits remaining in the **ATD** string. The **W** dial modifier also acts as a delimiter between the primary and secondary sections of the dial string, so that no additional **X** modifier is needed. Once either an **X** or a **W** has been parsed in the dial string, any additional **X** modifiers are ignored. Additional **W** modifiers cause Cisco MICA modems to wait for a dial tone.
,	Delay: Number of seconds in S8. Default is 2 seconds. The comma (**,**) dial modifier is treated as a silent DTMF tone for the duration of seconds specified in S8. The comma is acted on only after the call switching module (CSM) has made the transition to DTMF mode, which requires that it either follow an **X** or a **W** in the dial string, or that the T1/E1 be configured for DTMF signaling.

In the following example dial string, the portion of the string before the **X** is dialed for the given line type used in your configuration. All digits after the **X** generate the appropriate DTMF tones.

```
atdT5550101x,,567
```

Changing Configurations Manually in Integrated Microcom Modems

You can change the running configuration of an integrated modem by sending individual modem AT commands. Manageable Microcom modems have an out-of-band feature, which is used to poll modem statistics and send AT commands. The Cisco IOS software uses a direct connect session to transfer information through this out-of-band feature. To send AT commands to a Microcom modem, you must permit a direct connect session for a specified modem, open a direct connect session, send AT commands to a modem, and clear the directly connected session from the modem when you are finished.

Open a direct connect session by entering the **modem at-mode** *slot/port* command in privileged EXEC mode. From here, you can send AT commands directly from your terminal session window to the internal Microcom modems. Most incoming or outgoing calls on the modems are not interrupted when you open a direct connect session and send AT commands. However, some AT commands interrupt a call—for example, the **ATH** command, which hangs up a call. Open and close one direct connect session at a time. Note that multiple open sessions slow down modem performance.

Refer to the AT command set that came with your router for a complete list of AT commands that you can send to the modems.

For Microcom modems, you can clear or terminate an active directly connected session in two ways:

- Press **Ctrl-C** after sending all AT commands as instructed by the system when you enter AT command mode.
- Enter a second Telnet session and execute the **clear modem at-mode** *slot/port* EXEC command. This method is used for closing a directly connected session that may have been mistakenly left open by the first Telnet session.

The following example illustrates use of the modem commands.

AT Mode Example for Integrated Modems

To establish a direct connect session to an internal or integrated modem (existing inside the router), such as the connection required for Microcom modems in the Cisco AS5200 access server, open a directly connected session with the **modem at-mode** command and then send an AT command to the specified modem. For example, the following example sends the AT command **at%v** to modem 1/1:

```
AS5200# modem at-mode 1/1
You are now entering AT command mode on modem (slot 1 / port 1).
Please type CTRL-C to exit AT command mode.
at%v

MNP Class 10 V.34/V.FC Modem Rev 1.0/85

OK
at\s

IDLE           000:00:00
LAST DIAL

NET ADDR:      FFFFFFFFFFFF
MODEM HW: SA 2W United States
4 RTS 5 CTS 6 DSR - CD 20 DTR - RI
MODULATION     IDLE
MODEM BPS      28800  AT%G0
MODEM FLOW     OFF    AT\G0
MODEM MODE     AUT    AT\N3
V.23 OPR.      OFF    AT%F0
AUTO ANS.      ON     ATS0=1
SERIAL BPS     115200 AT%U0
BPS ADJUST     OFF    AT\J0
```

```
SPT BPS ADJ.    0       AT\W0
ANSWER MESSGS   ON      ATQ0
SERIAL FLOW     BHW     AT\Q3
PASS XON/XOFF   OFF     AT\X0
PARITY          8N      AT
```

The modem responds with "OK" when the AT command you send is received.

Configuring Leased-Line Support for Analog Modems

Analog modems on the NM-8AM and NM-16AM network modules in the Cisco 2600 and 3600 series routers provide two-wire leased-line support for enterprise customers who require point-to-point connections between locations and for enterprise customers with medium to high data transfer requirements without access to other technologies or with access to only low-grade phone lines.

This feature works only with leased lines that provide loop current. Each modem used must have an RJ-11 connection to the PSTN.

Several features enhance the analog modem software:

- 2-wire leased-line support.
- Modem speeds up to 33.6 kbps with support for all current analog modem protocols, compression, and error correction techniques.
- Power-on autoconnect and loopback testing.
- Support for the maximum number of leased-line users without data transmission loss at distances up to 2 to 5 km.
- In-band and out-of-band monitoring.
- Support on all Cisco 2600 and Cisco 3600 series platforms and upgradability using Cisco IOS software.
- Compatibility with other major leased-line modem vendors.

To configure this support, configure one modem AT command (**AT&L**) and two AT registers with the **modemcap entry** command for the appropriate leased lines.

For leased line configuration using the **AT&L**{**0** | **1** | **2**}command:

- **0**—Disables the leased line (enables switched line; default).
- **1**—Enables the leased line. The modem initiates a leased line when dial and answer commands (**ATD** and **ATA**) are issued.
- **2**—Enables the leased line. The modem goes off hook automatically after T57 number of seconds in:
  - Originate mode if ATS0 is 0.
  - Answer mode if ATS0 is not equal to 0.

The following AT registers can also be set:

- **AT:T57**—Number of seconds before going off hook in leased-line mode when the command **AT&L2** is used (defaults to 6).
- **AT:T79**—Number of autoretrains before the modem is disconnected (defaults to 3).

For more information about using the AT command set with the modems on the NM-8AM and NM-16AM network modules in the Cisco 2600 and 3600 series routers, search Cisco.com for the publication *AT Command Set and Register Summary for Analog Modem Network Modules.*

To configure a modem for leased-line operation, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# modemcap entry modem-type-name:AA=S0=0&L2`	Sets the modemcap for leased-line operation for the originating modem.
Step 2	`Router(config)# modemcap entry modem-type-name:AA=S0=1&L2`	Sets the modemcap for leased-line operation for the answering modem.

The **show modemcap** command lists all the predefined modem types and any user-defined modemcaps that are currently configured on the router:

- If the leased line has been configured, the modemcap information will be available.
- If the leased line has not been configured, only the predefined modem types will be displayed.

The important setting for leased-line support is what is defined in the modemcap as the key configuration item and its application to the leased line. Consider the following command strings:

```
modemcap entry micro_LL_orig:AA=S0=0&L2
modemcap entry micro_LL_ans:AA=S0=1&L2
```

AA stands for autoanswer:

- The answering modem AA register is set to 1 (AA=S0=1) so that autoanswer is “on”.
- The originating modem AA register is set to 0 (AA=S0=0) so that autoanswer is “off”.

If the AA feature is used, both the originating and answering modem must be put into leased-line mode with the **&L2** AT command.

In the examples, the micro_LL_orig and micro_LL_ans strings are arbitrary text descriptions.

Note For the **modemcap entry** command, one of the predefined modem types may be used or a completely user-defined modemcap may be created. For leased line, no new modem type was added. Users may create their own modemcaps for leased-line functionality.

To configure the modem for leased-line operation, use the **modemcap entry** command. For each connection, each modem must be configured as an originator or answerer.

The following example shows modemcaps for a leased-line originator and answerer and their application to specific ports:

```
modemcap entry micro_LL_orig:AA=S0=0&L2
modemcap entry micro_LL_ans:AA=S0=1&L2
line 73
 no exec
 modem InOut
 modem autoconfigure type micro_LL_ans
 transport input all
line 74
 no exec
 modem InOut
 modem autoconfigure type micro_LL_orig
 transport input all
```

Note When Multilink PPP (MLP) is configured on a dialer interface, the dialer configuration has a default value of 2 minutes for dialer idle timeout. For leased-line connections, set the dialer idle timeout to infinity by adding **dialer idle-timeout 0** to the configuration.

Verifying the Analog Leased-Line Configuration

The following information is important for verifying or troubleshooting your configuration. The **show modem log** command displays the progress of leased-line connections. Here is an example log for a leased-line answerer. Note the "LL Answering" state and "LL Answer" in the "Direction" field of the connection report:

```
00:44:03.884 DTR set high
00:44:02.888 Modem enabled
00:43:57.732 Modem disabled
00:43:52.476 Modem State:LL Answering
00:43:52.476 CSM:event-MODEM_STARTING_CONNECT New
State-CSM_CONNECT_INITIATED_STATE
00:43:51.112 Modem State:Waiting for Carrier
00:43:43.308 Modem State:Connected
00:43:42.304 Connection:TX/RX Speed = 33600/33600,
Modulation = V34
Direction = LL Answer, Protocol = MNP, Compression =
V42bis
00:43:42.304 CSM:event-MODEM_CONNECTED New
State-CONNECTED_STATE
00:43:42.300 RS232:noCTS* DSR* DCD* noRI noRxBREAK
TxBREAK*
00:43:41.892 PPP mode active
00:43:41.892 Modem enabled
00:43:39.888 PPP escape maps set:TX map=00000000 RX
map=FFFFFFFF
00:43:39.724 PPP escape maps set:TX map=00000000 RX
map=000A0000
00:43:34.444 RS232:CTS* DSR DCD noRI noRxBREAK TxBREAK
00:43:11.716 Modem Analog Report:TX = -20, RX = -34,
Signal to noise = 61
```

Cisco 2600 and 3600 Series Analog Modem Leased-Line Support Examples

In the following examples, one Cisco 3620 router and one Cisco 3640 router are connected back-to-back using leased lines. The Cisco 3620 router has the originating configuration, and the Cisco 3640 router has the answering configuration.

In the dialer interface configuration, the **dialer idle-timeout 0** command is added to set the dialer idle timeout to be infinity. Otherwise the leased line will go down and up every 2 minutes because the default dialer interface idle timeout is 2 minutes.

Note Except for passwords and logins, the Cisco IOS command-line interface (CLI) is case-insensitive. For this document, an uppercase "L" has been used in the command examples to avoid confusion with the numeral "1".

Leased-Line Originating Configuration

```
version 12.1
service timestamps debug uptime
service timestamps log uptime
!
```

```
modemcap entry micro_LL_orig:AA=S0=0&L2
modemcap entry micro_LL_ans:AA=S0=1&L2
!
interface Async33
  no ip address
encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 dialer in-band
 dialer pool-member 1
 async default routing
 async dynamic routing
 async mode dedicated
 no peer default ip address
 no fair-queue
 no cdp enable
 ppp direction callout
 ppp multilink
!
interface Dialer1
 ip address 10.1.24.1 255.255.255.0
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 dialer remote-name sara40
 dialer pool 1
 dialer idle-timeout 0
 dialer max-call 4096
 no cdp enable
 ppp direction callout
 ppp multilink
!
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
 transport input none
 line 33
 no exec
 modem InOut
 modem autoconfigure type micro_LL_orig
 transport input all
line aux 0
 exec-timeout 0 0
line vty 0 4
 exec-timeout 0 0
!
end
```

Leased-Line Answering Configuration

```
version 12.1
service timestamps debug uptime
service timestamps log uptime
!
modemcap entry micro_LL_orig:AA=S0=0&L2
modemcap entry micro_LL_ans:AA=S0=1&L2
!
interface Async73
 no ip address
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 dialer in-band
```

```
 dialer pool-member 1
 async default routing
 async dynamic routing
 async mode dedicated
 no peer default ip address
 no fair-queue
 no cdp enable
 ppp direction callout
 ppp multilink
!
interface Dialer1
 ip address 10.1.24.2 255.255.255.0
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 load-interval 30
 dialer remote-name sara20
 dialer pool 1
 dialer idle-timeout 0
 dialer load-threshold 1 either
 dialer max-call 4096
 no cdp enable
 ppp direction callout
 ppp multilink
!
dialer-list 1 protocol ip permit
line con 0
 exec-timeout 0 0
 transport input none
line 73
 no exec
 modem InOut
 modem autoconfigure type micro_LL_ans
 transport input all
line aux 0
 transport input all
 flowcontrol hardware
line vty 0 4
 exec-timeout 0 0
!
end
```

Configuring Modem Pooling

Modem pooling allows you to control which modem a call connects to, on the basis of dialed number identification service (DNIS). When modem pooling is not used, incoming and outgoing calls are arbitrarily assigned to modems. For example, consider a Cisco AS5300 access server loaded with a 4-port ISDN PRI card. After an analog modem call comes into the first PRI trunk, the call is greeted by a general pool of B channels and a general pool of modems. Any B channel can be connected to any modem in the access server. A random assignment takes place. Modem resources cannot be controlled.

Modem pooling assigns physical modems to a single DNIS. It enables you to create pools of physical modems in one access server, assign a unique DNIS to each modem pool, and set maximum simultaneous connect limits.

This feature is used for physically partitioning or virtually partitioning modems inside one network access server.

Modem pooling offers these benefits:

- A certain number of modem ports can be guaranteed per DNIS.
- Maximum simultaneous connection limits can be set for each DNIS.

The following restrictions apply:

- Modem pooling is not a solution for large-scale dial access. It cannot be used to create virtual modem pools across multiple access servers that are connected. Modem pooling is physically restricted to one access server.
- MICA and Microcom technology modems support modem pooling. However, only MICA modems support modem pooling for CT1 and CE1 configurations using CAS. To use modem pooling with CT1 or CE1 connections, you must reserve at least two modems in the default modem pool. These reserved modems decode DNIS before handing off calls to the modems assigned to modem pools.

  If you see many call failures appearing on the access server, try assigning more modems to the default pool. Use the **show modem** and **show modem summary** EXEC commands to display the modem call failure and success ratio.
- No MIBs support modem pooling.
- The same DNIS cannot exist in more than one modem pool.

Modem pooling is supported on the Cisco AS5300 access servers. To configure and manage modems, perform the tasks in the following sections; all tasks are optional and depend upon the needs of your system.

- Creating a Modem Pool (Required)
- Verifying Modem Pool Configuration (As required)

Creating a Modem Pool

You must first decide to physically partition or virtually partition your modems. For more information, see the previous section, "Configuring Modem Pooling." After you have made this decision, create a modem pool for a dial-in service or specific customer by using the following commands beginning in global configuration mode.

	Command	Purpose
Step 1	`Router(config)# modem-pool` *name*	Creates a modem pool and assigns it a name, and starts modem pool configuration mode.
Step 2	`Router(config-modem-pool)# pool-range` *number-number*	Assigns a range of modems to the pool. A hyphen (**-**) is required between the two numbers. The range of modems you can choose from is equivalent to the number of modems in your access server that are not currently associated with another modem pool.
Step 3	`Router(config-modem-pool)# called-number` *number* [**max-conn** *number*]	Assigns the DNIS to be used for this modem pool. The **max-conn** option specifies the maximum number of simultaneous connections allowed for this DNIS. If you do not specify a **max-conn** value, the default (total number of modems in the pool) is used.[1]
Step 4	`Router(config-modem-pool)# Ctrl-Z`	Returns to EXEC mode.

	Command	Purpose
Step 5	`Router# show configuration`	Displays the running configuration to verify the modem pool settings. Make changes accordingly.
Step 6	`Router# copy running-config startup-config`	Saves the running configuration to the startup configuration.

1. The DNIS string can have an integer x to indicate a "don't care" digit for that position, for example, 555010x.

Note If you have active modem calls on the access server before using modem pooling, modem pooling gracefully applies itself to the access server. Modem pooling first waits for active calls to hang up before assigning modems to modem pools and directing calls according to DNIS.

Verifying Modem Pool Configuration

To verify the modem configuration, enter the **show modem-pool** command to display the configuration. This command displays the structure and activity status for all the modem pools in the access server. See Table 12 for a description of each display field.

```
Router# show modem-pool

modem-pool: System-def-Mpool
modems in pool: 0   active conn: 0
0 no free modems in pool

modem-pool: v90service
modems in pool: 48  active conn: 46
 8 no free modems in pool
 called_party_number: 1234
   max conn allowed: 48, active conn: 46
   8 max-conn exceeded, 8 no free modems in pool

modem-pool: v34service
modems in pool: 48  active conn: 35
 0 no free modems in pool
 called_party_number: 5678
   max conn allowed: 48, active conn: 35
   0 max-conn exceeded, 0 no free modems in pool
```

Table 12 show modem-pool Field Descriptions

Field	Description
modem-pool	Name of the modem pool. In the previous example, there are three modem pools configured: System-def-Mpool, v90service, and v34service. To set the modem pool name, refer to the **modem-pool** command. All the modems not assigned to a modem pool are automatically assigned to the system default pool (displayed as System-def-Mpool).
modems in pool	Number of modems assigned to the modem pool. To assign modems to a pool, refer to the display and descriptions for the **pool-range** command.

Table 12 show modem-pool Field Descriptions (continued)

Field	Description
active conn	Number of simultaneous active connections for the specified modem pool or called party DNIS number.
no free modems in pool	Number of times incoming calls were rejected because there were no more free modems in the pool to accept the call.
called_party_number	Specified called party DNIS number. This is the number that the remote clients use to dial in to the access server. You can have more than one DNIS number per modem pool. To set the DNIS number, refer to the description for the **called-number** command.
max conn allowed	Maximum number of modems that a called party DNIS number can use, which is an overflow protection measure. To set this feature, refer to the description for the **called-number** command.
max-conn exceeded	Number of times an incoming call using this called party DNIS number was rejected because the **max-conn** *number* parameter specified by the **called-number** command was exceeded.

For modem pool configuration examples, see the section "Physical Partitioning with Dial-In and Dial-Out Scenario" later in this chapter.

Check the following if you are having trouble operating your modem:

- Make sure you have not configured the same DNIS for multiple pools.
- Make sure you have not placed the same modem in multiple pools.

Note Modem pools that use MICA or Microcom modems support incoming analog calls over ISDN PRI. However, only MICA modems support modem pooling for T1 and E1 configurations with CAS.

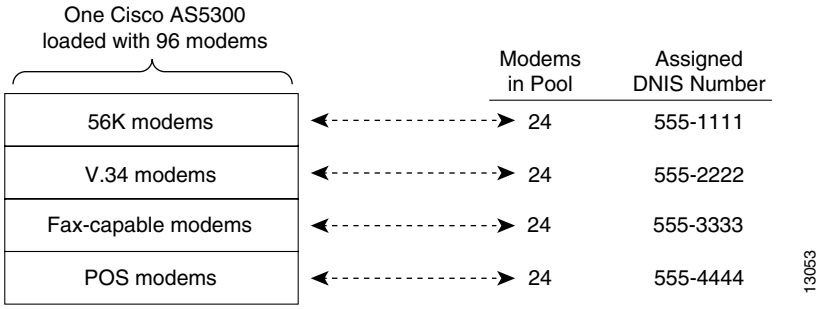
Configuring Physical Partitioning

You can either physically partition or virtually partition your modems to enable different dial-in and dial-out services. This section provides information about the following optional tasks:

- Creating a Physical Partition, page 85
- Physical Partitioning with Dial-In and Dial-Out Scenario, page 87

Physical partitioning uses one access server to function as multiple access servers loaded with different types of modem services (for example, V.34 modems, fax-capable modems, and point-of-sale (POS) modems). Each modem service is part of one physical modem pool and is assigned a unique DNIS number. (See Figure 19.)

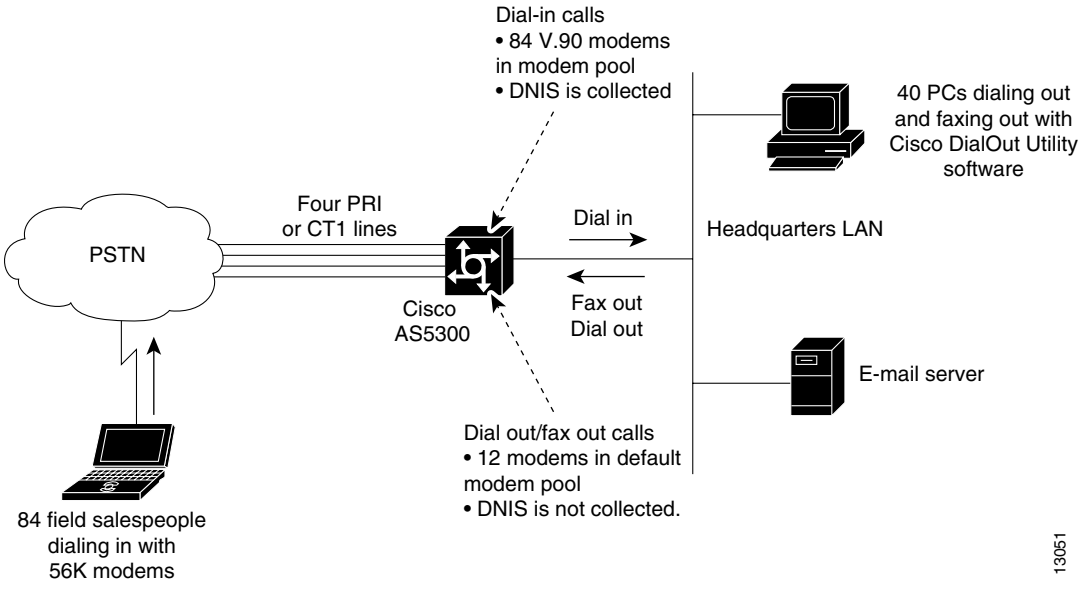
Figure 19 Modem Pooling Using Physical Partitioning



Physical partitioning can also be used to set up an access server for bidirectional dial access. (See Figure 20.)

Figure 20 shows one Cisco AS5300 access server loaded with 96 MICA modems and configured with 2 modem pools. One modem pool has 84 modems and collects DNIS. This pool is shared by 400 salespeople who remotely download e-mail from headquarters. The other modem pool contains 12 fax-capable modems and does not collect DNIS. This pool is shared by 40 employees using PCs on a LAN. Each time an outbound call is initiated by a PC, a modem on the Cisco AS5300 access server is seized and used to fax out or dial out. Not configuring DNIS support in the fax-out modem pool protects the pool from being used by the calls coming in from the field. Regardless of how many salespeople are dialing in or which telephone number they use, the fax-out and dial-out modem pool will always be reserved for the PCs connected to the LAN.

Figure 20 Modem Pooling Used for Bidirectional Dialing



Creating a Physical Partition

The following task creates one V.34 modem pool and one 56K modem pool on a Cisco AS5200. Each modem pool is configured with its own DNIS. Depending on which DNIS the remote clients dial, they connect to a 56K MICA modem or a V.34 Microcom modem.

The following hardware configuration is used on the Cisco AS5200 access server:

- One 2-port T1 PRI card
- One 48-port card containing four 6-port MICA 56K modem modules and two 12-port Microcom V.34 modem modules

To configure basic physical partitioning, perform the following steps:

Step 1 Enter global configuration mode:

```
Router# configure terminal
Router(config)#
```

Step 2 Create the modem pool for the 56K MICA modem services using the **modem-pool** *name* command. The modem pool is called 56kservices, which spans four 6-port MICA 56K modem modules.

```
Router(config)# modem-pool 56kservices
Router(config-modem-pool)#
```

Note The router is in modem pool configuration mode after the prompt changes from Router(config)# to Router(config-modem-pool)#.

Step 3 Assign a range of modems to the modem pool using the **pool-range** *number-number* command. Because all the 56K MICA technologies modems are seated in slot 1, they are assigned TTY line numbers 1 to 24. Use the **show line** EXEC command to determine the TTY line numbering scheme for your access server.

```
Router(config-modem-pool)# pool-range 1-24
```

Step 4 Assign a DNIS to the modem pool using the **called-number** *number* [**max-conn** *number*] command. This example uses the DNIS 5550101 to connect to the 56K modems. The maximum simultaneous connection limit is set to 24. The 25th user who dials 5550101 gets a busy signal.

```
Router(config-modem-pool)# called-number 5550101 max-conn 24
```

Step 5 Return to EXEC mode by entering **Ctrl-Z**. Next, display the modem pool configuration using the **show modem-pool** command. In the following example, 56K modems are in the modem pool called 56kservices. The remaining 24 V.34 Microcom modems are still in the default system pool.

```
Router(config-modem-pool)# ^Z
Router# show modem-pool

modem-pool: System-def-Mpool
modems in pool: 24   active conn: 0
0 no free modems in pool

modem-pool: 56kservices
modems in pool: 24  active conn: 0
 0 no free modems in pool
 called_party_number: 5550101
   max conn allowed: 24, active conn: 0
   0 max-conn exceeded, 0 no free modems in pool
```

Step 6 Create the modem pool for the Microcom physical partition. After the configuration is complete, the **show modem-pool** command shows that there are no remaining modems in the system default modem pool.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# modem-pool v34services
```

```
Router(config-modem-pool)# pool-range 25-48
Router(config-modem-pool)# called-number 5550202 max-conn 24
Router(config-modem-pool)# ^Z
Router# show modem-pool

modem-pool: System-def-Mpool
modems in pool: 0   active conn: 0
0 no free modems in pool

modem-pool: 56kservices
modems in pool: 48  active conn: 0
 0 no free modems in pool
 called_party_number: 5550101
 max conn allowed: 48, active conn: 0
 0 max-conn exceeded, 0 no free modems in pool

modem-pool: v34services
modems in pool: 48  active conn: 0
 0 no free modems in pool
 called_party_number: 5550202
 max conn allowed: 48, active conn: 0
 0 max-conn exceeded, 0 no free modems in pool

Router# copy running-config startup-config
```

Physical Partitioning with Dial-In and Dial-Out Scenario

The following is a bidirectional dial scenario using a Cisco AS5300 access server. Two modem pools are configured. One modem pool contains 84 56K MICA modems, which is shared by 400 remote salespeople who dial in to headquarters. The other modem pool contains 12 fax-capable modems, which are shared by 40 employees who dial out of the headquarters LAN using the Cisco DialOut Utility software. See Figure 20 for the network topology.

The following hardware configuration is used on the Cisco AS5300:

- One 4-port T1 PRI card
- Two 48-port cards containing fourteen 6-port MICA 56K modem modules and two 6-port MICA fax-capable modem modules

To configure physical partitioning with dial-in and dial-out capability, perform the following steps:

Step 1 Create the 56K modem pool for the 400 remote salespeople. This modem pool contains 84 modems, which are reserved for the dial-in calls. To get access, the salespeople dial the DNIS 5550303. The total number of simultaneous calls is limited to 84. The 85th call and those above it are rejected. The **modem dialin** line configuration command is used to prevent modems 1 to 84 from dialing out.

```
Router# configure terminal
Router(config)# modem-pool 56ksalesfolks
Router(config-modem-pool)# pool-range 1-84
Router(config-modem-pool)# called-number 5550303 max-conn 84
Router(config-modem-pool)# exit
Router(config)# line 1 84
Router(config-line)# modem dialin
Router(config-line)# transport input all
Router(config-line)# rotary 1
Router(config-line)# autoselect ppp
Router(config-line)# exit
Router(config)#
```

Step 2 Create the dial-out/fax-out modem pool for the 40 local employees connected to the headquarters LAN. This modem pool contains 12 fax-capable MICA modems. No DNIS is assigned to the pool. Because lines 85 to 96 are used for the dial-out and fax-out modem services, the asynchronous lines are configured for reverse Telnet. This configuration is needed for the Telnet extensions to work with the dial-out application, which is installed on the LAN PCs.

```
Router(config)# modem-pool dialoutfolks
Router(config-modem-pool)# pool-range 85-96
Router(config-modem-pool)# exit
Router(config)# line 85-96
Router(config-line)# refuse-message z [!NMM!] No Modems Available z
Router(config-line)# exec-timeout 0 0
Router(config-line)# autoselect during-login
Router(config-line)# autoselect ppp
Router(config-line)# modem inout
Router(config-line)# rotary 1
Router(config-line)# transport preferred telnet
Router(config-line)# transport input all
Router(config-line)# exit
Router(config)#
```

Step 3 Configure the group asynchronous interface, which assigns core protocol characteristics to all the asynchronous interfaces in the system. Regardless of the direction that the modems are dialing, all modems in the access server leverage this group asynchronous configuration.

```
Router(config)# interface group-async 1
Router(config-if)# ip unnumbered ethernet 0
Router(config-if)# encapsulation ppp
Router(config-if)# async mode interactive
Router(config-if)# ppp authentication chap pap paplocal
Router(config-if)# peer default ip address pool bidir_dial_pool
Router(config-if)# no cdp enable
Router(config-if)# no ip mroute cache
Router(config-if)# no ip route cache
Router(config-if)# async dynamic routing
Router(config-if)# async dynamic address
Router(config-if)# group range 1-96
Building configuration...
Router(config-if)# exit
```

Step 4 Create an IP address pool for all the dial-in clients and dial-out clients. Both types of clients borrow addresses from this shared pool.

```
Router(config)# ip local pool bidir_dial_pool 10.4.1.1 10.4.1.96
Router(config)# ^z
Router# copy running-config startup-config
```

Step 5 (Optional) If you are using CiscoSecure AAA and a remote TACACS server, include the following security statements on the access server:

```
Router(config)# aaa new-model
Router(config)# aaa authentication login default tacacs+
Router(config)# aaa authentication login noaaa local
Router(config)# aaa authentication login logintac tacacs+
Router(config)# aaa authentication ppp ppptac tacacs+
Router(config)# aaa authentication ppp paplocal local
Router(config)# aaa authorization exec tacacs+
Router(config)# aaa authorization network tacacs+
Router(config)# aaa authorization reverse-access tacacs+
Router(config)# aaa accounting exec start-stop tacacs+
Router(config)# aaa accounting network start-stop tacacs+
Router(config)# aaa accounting update newinfo
Router(config)# enable password cisco
```

You should also include the host name, timeout interval, and authentication key:

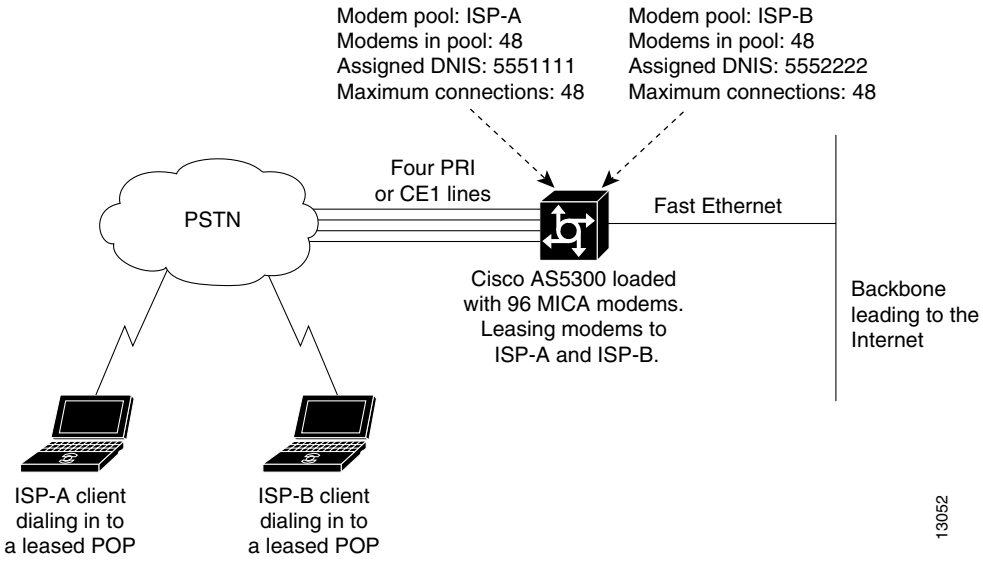
```
Router(config)# tacacs-server host 10.4.1.10
Router(config)# tacacs-server timeout 20
Router(config)# tacacs-server key nas1
```

Configuring Virtual Partitioning

Virtual partitioning creates one large modem pool on one access server, but assigns different DNIS numbers to different customers. Each incoming DNIS consumes resources from the same modem pool, but a maximum connect option is set for each DNIS.

Figure 21 shows two Internet service provider (ISP) customers who are leasing modems from another service provider. Each ISP is assigned its own DNIS number and range of modems. Each ISP is guaranteed a certain number of physical modem ports for simultaneous connections. After an ISP uses up all the modems assigned to its DNIS, a busy signal is issued.

Figure 21 Modem Pooling Using Virtual Partitioning



Virtual partitioning essentially resells modem banks to customers, such as a small-sized ISP. However, remember that modem pooling is a single-chassis solution, not a multichassis solution. Modem pooling is not a solution for reselling ports on a large-scale basis.

The following procedure creates one modem pool on a Cisco AS5300 access server for two ISP customers. The shared modem pool is called isp56kpool. However, both ISP customers are assigned different DNIS numbers and are limited to a maximum number of simultaneous connections.

See Figure 21 for the network topology.

The following hardware configuration is used on the Cisco AS5300 access server:

- One 4-port T1 PRI card
- Two 48-port cards containing sixteen 6-port MICA 56K modem modules

To configure virtual partitioning, perform the following steps:

Step 1 Enter global configuration mode:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Step 2 Create the shared modem pool for the 56K MICA modem services. This modem pool is called isp56kpool, which spans sixteen 6-port MICA 56K modem modules.

```
Router(config)# modem-pool isp56kpool
Router(config-modem-pool)#
```

Step 3 Assign all the modems to the modem pool using the **pool-range** *number-number* command. Use the **show line** EXEC command to determine your TTY line numbering scheme.

```
Router(config-modem-pool)# pool-range 1-96
```

Step 4 Assign a unique DNIS to each ISP customer using the **called-number** *number* [**max-conn** *number*] command. In this example, the **max-conn** *number* option limits each ISP to 48 simultaneous connections. The 49th user to dial either DNIS will get a busy signal.

```
Router(config-modem-pool)# called-number 5550101 max-conn 48
Router(config-modem-pool)# called-number 5550202 max-conn 48
```

Step 5 Return to EXEC mode by entering a **Ctrl-Z** sequence. Next, display the modem pool configuration using the **show modem-pool** command. In the following example, all the 56K modems are in the isp56kpool modem pool. The output also shows two DNIS numbers configured: 5550101 and 5550202.

```
Router(config-modem-pool)# ^Z
Router# show modem-pool
modem-pool: System-def-Mpool
modems in pool: 0   active conn: 0
0 no free modems in pool

modem-pool: isp56kpool
modems in pool: 96  active conn: 0
 0 no free modems in pool
 called_party_number: 5550101
   max conn allowed: 48, active conn: 0
   0 max-conn exceeded, 0 no free modems in pool
 called_party_number: 5550202
   max conn allowed: 48, active conn: 0
   0 max-conn exceeded, 0 no free modems in pool
Router# copy running-config startup-config
```

Configuring Call Tracker

The Call Tracker feature captures detailed statistics on the status and progress of active calls and retains historical data for disconnected call sessions. Call Tracker collects session information such as call states and resources, traffic statistics, total bytes transmitted and received, user IP address, and disconnect reason. This data is maintained within the Call Tracker database tables, which are accessible through the Simple Network Management Protocol (SNMP), the CLI, or syslog.

Note The calltracker command, providing Call Tracker services, is supported for dial calls but not voice. Calltracker is supported for dial calls on 5x platforms (5300, 5350, 5400, 5800, and 5850).

Call Tracker is notified of applicable call events by related subsystems such as ISDN, PPP, CSM, Modem, EXEC, or TCP-Clear. SNMP traps are generated at the start of each call, when an entry is created in the active table, and at the end of each call, when an entry is created in the history table. Call Record syslogs are available through configuration that will generate detailed information records for all call terminations. This information can be sent to syslog servers for permanent storage and future analysis.

Additionally, the status and diagnostic data that is routinely collected from MICA modems is expanded to include new link statistics for active calls, such as the attempted transmit and receive rates, the maximum and minimum transmit and receive rates, and locally and remotely issued retrains and speedshift counters. For more detailed information on Call Tracker logs, refer to the TAC Tech Notes document, *Understanding Call Tracker Outputs*, at the following URL: http://www.cisco.com/warp/public/471/calltracker_view.html

To configure Call Tracker, perform the following steps:

	Command	Purpose
Step 1	`Router(config)# calltracker enable`	Enables Call Tracker.
Step 2	`Router(config)# calltracker call-record {terse|verbose} [quiet]`	Enables Call Tracker syslog support for generating detailed Call Records.
Step 3	`Router(config)# calltracker history max-size number`	Sets the maximum number of call entries to store in the Call Tracker history table.
Step 4	`Router(config)# calltracker history retain-mins minutes`	Sets the number of minutes for which calls are stored in the Call Tracker history table.
Step 5	`Router(config)# snmp-server packetsize byte-count`	Sets the maximum packet size allowed for SNMP server requests and replies.
Step 6	`Router(config)# snmp-server queue-length length`	Sets the queue length for SNMP traps.
Step 7	`Router(config)# snmp-server enable traps calltracker`	Enables Call Tracker to send traps whenever a call starts or ends.
Step 8	`Router(config)# snmp-server host host community-string calltracker`	Specifies the name or Internet address of the host to send Call Tracker traps.

Verifying Call Tracker

To verify the operation of Call Tracker, use the the following command in EXEC mode:

Command	Purpose
`Router# show call calltracker summary`	Verifies the Call Tracker configuration and current status.

Enabling Call Tracker

The following example shows how to enable the Call Tracker feature:

```
calltracker enable
```

```
calltracker call-record terse
calltracker history max-size 50
calltracker history retain-mins 5000
!
snmp-server engineID local 0012345
snmp-server community public RW
snmp-server community private RW
snmp-server community wxyz123 view v1default RO
snmp-server trap-source FastEthernet0
snmp-server packetsize 17940
snmp-server queue-length 200
snmp-server location SanJose
snmp-server contact Bob
snmp-server enable traps snmp
snmp-server enable traps calltracker
snmp-server enable traps isdn call-information
snmp-server enable traps hsrp
snmp-server enable traps config
snmp-server enable traps entity
snmp-server enable traps envmon
snmp-server enable traps bgp
snmp-server enable traps ipmulticast-heartbeat
snmp-server enable traps rsvp
snmp-server enable traps frame-relay
snmp-server enable traps rtr
snmp-server enable traps syslog
snmp-server enable traps dlsw
snmp-server enable traps dial
snmp-server enable traps dsp card-status
snmp-server enable traps voice poor-qov
snmp-server host 10.255.255.255 wxyz123
snmp-server host 10.0.0.0 xxxyyy calltracker
!
radius-server host 172.16.0.0 auth-port 1645 acct-port 1646 non-standard
radius-server key xyz
!
```

Configuring Polling of Link Statistics on MICA Modems

The status and diagnostic data that is routinely collected from MICA modems is expanded to include new link statistics for active calls, such as the attempted transmit and receive rates, the maximum and minimum transmit and receive rates, and locally and remotely issued retrains and speedshift counters. This connection data is polled from the modem at user-defined intervals and passed to Call Tracker.

To poll modem link statistics, use the following command in global configuration mode:

Command	Purpose
`Router(config)# modem link-info poll time` *seconds*	Sets the polling interval at which link statistics for active calls are retrieved from the modem.

Note The **modem link-info poll time** command consumes a substantial amount of memory, approximately 500 bytes for each MICA modem call. Use this command only if you require the specific data that it collects; for instance, if you have enabled Call Tracker on your access server.

Configuring MICA In-Band Framing Mode Control Messages

Dial-in Internet connections typically start in character mode to allow the user to log in and select a preferred service. When Cisco IOS software determines that the user wants a framed interface protocol during the call, such as PPP or SLIP, commands are sent to the MICA modem so that it will provide hardware assistance with the framing. This hardware assistance reduces the Cisco IOS processing load. To avoid loss or misinterpretation of framed data during the transition, issue these commands at precise times with respect to the data being sent and received.

MICA modem framing commands can be sent in the data stream itself, which greatly simplifies Cisco IOS tasks in achieving precision timing. For PPP connections, the common way for modems to connect to the Internet, total connect time might typically be improved by 2 to 3 seconds. This functionality reduces timeouts during PPP startup and reduces startup time. If an ASCII banner is sent just before PPP startup, this feature eliminates problems with banner corruption such as truncation and extraneous characters, thus improving the performance of terminal equipment.

In earlier software, the modem interface timing rules were not well understood and were difficult or impossible to implement using the separate command interface of the modem. The practical result is that the MICA in-band framing mode reduces the number of timeouts during PPP startup, and thus reduces startup time. MICA in-band framing is supported on MICA modems in Cisco AS5300 and Cisco AS5800 access servers.

To configure the MICA in-band framing mode control messages, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line` *line-number* [*ending-line-number*]	Specifies the number of modem lines to configure and enters line configuration mode. If a range is entered, it must be equal to the number of modems in the router.
Step 2	`Router(config-line)# no flush-at-activation`	Improves PPP and SLIP startup. Normally a router avoids line and modem noise by clearing the initial data received within the first one or two seconds. However, when the autoselect PPP feature is configured, the router flushes characters initially received and then waits for more traffic. This flush causes timeout problems with applications that send only one carriage return.

The Cisco IOS software offers additional interface commands that can be set to control modem interface timing. Refer to the Cisco IOS command references for more information about the interface commands described in the following paragraphs.

When a link goes down and comes back up before the timer set by the **carrier-delay** command expires, the down state is effectively filtered, and the rest of the software on the switch is not aware that a link-down event occurred. Therefore, a large carrier delay timer results in fewer link-up and link-down events being detected. On the other hand, setting the carrier delay time to 0 means that every link-up and link-down event is detected.

When the link protocol goes down (because of loss of synchronization, for example), the interface hardware is reset and the data terminal ready (DTR) signal is held inactive for at least the specified interval. Setting the **pulse-time** command enable pulsing DTR signal intervals on serial interfaces, and is useful for handling encrypting or other similar devices that toggle the DTR signal to resynchronize.

Use the **modem dtr-delay** command to reduce the time that a DTR signal is held down after an asynchronous line clears and before the DTR signal is raised again to accept new calls. Incoming calls may be rejected in heavily loaded systems, even when modems are unused because the default DTR hold-down interval may be too long. The **modem dtr-delay** command is designed for lines used for an unframed asynchronous session such as Telnet. Lines used for a framed asynchronous session such as PPP should use the **pulse-time** interface command.

Enabling Modem Polling

The following example enables modem status polling through the out-of-band feature, which is associated to line 1:

```
Router# configure terminal
Router(config)# line 1
Router(config-line)# modem status-poll
```

Setting Modem Poll Intervals

The following example sets the time interval between polls to 10 seconds using the **modem poll time global** configuration command:

```
Router# configure terminal
Router(config)# modem poll time 10
```

Setting Modem Poll Retry

The following example configures the server to attempt to retrieve statistics from a local modem up to five times before discontinuing the polling effort:

```
Router# configure terminal
Router(config)# modem poll retry 5
```

Collecting Modem Statistics

Depending upon your modem type, the Cisco IOS software provides several **show** EXEC commands that allow you to display or poll various modem statistics. See Table 7 and Table 8 to find the **show** EXEC command appropriate for your modem type and the task you want to perform.

Logging EIA/TIA Events

To facilitate meaningful analysis of the modem log, turn the storage of specific types of EIA/TIA events on or off. To activate or inactivate the storage of a specific type of EIA/TIA modem event for a specific line or set of lines, use either of the following commands in line configuration mode, as needed:

Command	Purpose
`Router(config-line)# modem log {cts | dcd | dsr | dtr | ri | rs323 | rts | tst}`	Configures the types of EIA/TIA events that are stored in the modem log. The default setting stores no EIA/TIA events.
or	
`Router(config-line)# no modem log {cts | dcd | dsr | dtr | ri | rs323 | rts | tst}`	Turns off the logging of a specific type of EIA/TIA event.

Configuring a Microcom Modem to Poll for Statistics

Manageable Microcom modems have an out-of-band feature, which is used for polling modem statistics. To configure the system to poll for modem statistics, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# modem poll time seconds`	Specifies the number of seconds between statistical modem polling for Microcom modems. The default is 12 seconds. The configuration range is from 2 to 120 seconds.
Step 2	`Router(config)# modem poll retry number`	Sets the maximum number of polling attempts to Microcom modems. The default is three polling attempts. The configuration range is from 0 to 10 attempts.[1]
Step 3	`Router(config)# modem status-poll`	Polls for status and statistics for a Microcom modem through the modem's out-of-band feature.
Step 4	`Router(config)# modem buffer-size number`	Defines the number of modem events that each modem is able to store. The default is 100 events for each modem. Use the **show modem log** command to display modem events.

1. If the number of attempts to retrieve modem status or statistics exceeds the number you define, the out-of-band feature is removed from operation. In this case, you must reset the modem hardware using the **clear modem** command.

Troubleshooting Using a Back-to-Back Modem Test Procedure

You can manually isolate an internal back-to-back connection and data transfer between two modems for focused troubleshooting purposes. For example, if mobile users cannot dial in to modem 2/5 (which is the sixth modem port on the modem board in the second chassis slot), attempt a back-to-back test with modem 2/5 and a modem known to be functioning, such as modem 2/6. You might need to enable this command on several different combinations of modems to determine which one is not functioning properly. A pair of operable modems connect and complete sending data in both directions. An operable modem and an inoperable modem do not connect with each other.

To perform the modem test procedure, enter the **test modem back-to-back** *first-slot/port second-slot/port* command, as follows:

Step 1 Perform a back-to-back modem test between two normal functioning modems. This example shows a successful connection between modem 1/1 and modem 1/0, which verifies normal operating conditions between these two modems:

```
Router# test modem back-to-back 1/1 1/0
Repetitions (of 10-byte packets) [1]: 10
Router#
%MODEM-5-B2BCONNECT: Modems (1/1) and (1/0) connected in back-to-back test: CONN
ECT9600/REL-MNP
%MODEM-5-B2BMODEMS: Modems (1/0) and (1/1) completed back-to-back test: success/
packets = 20/20
```

After you enter the **test modem back-to-back** command, you must define the number of packets sent between modems at the Repetitions prompt. The ideal range of packets to send and receive is from 1 to 100. The default is 1 packet that is 10 bytes large. The response message (for example, "success/packets = 20/20") tells you how many packets were sent in *both* directions compared to the total number of packets attempted to be sent in both directions. Because the software reports the packet total in both directions, the reported numbers are *two times* the number you originally specify.

When a known good modem is tested against a known bad modem, the back-to-back modem test fails. In the following example, modem 1/3 is suspected or proven to be inoperable or bad:

```
Router# test modem back-to-back 1/1 1/3
Repetitions (of 10-byte packets) [1]: 10
Router#
%MODEM-5-BADMODEMS: Modems (1/3) and (1/1) failed back-to-back test: NOCARRIER
```

Step 2 You would need to manually mark modem 1/3 as an inoperable or bad modem. You mark the bad modem by determining which line number corresponds with the modem. Use the **show modem 1/3** EXEC command to verify that TTY line number 4 (shown as `TTY4`) is used for modem 1/3:

```
Router# show modem 1/3
Mdm  Typ     Status     Tx/Rx     G  Duration  TX  RX  RTS  CTS  DSR  DCD  DTR
1/3  V34     Idle    28800/28800  0  00:00:00           x    x    x    x    x

Modem 1/3, Microcom MNP10 V34 Modem (Managed), TTY4
Firmware (Boot) Rev: 1.0(23) (1.0(5))
Modem config: Incoming and Outgoing
Protocol: reliable/MNP, Compression: V42bis
Management port config: Status polling and AT session
Management port status: Status polling and AT session
TX signals: -15 dBm, RX signals: -17 dBm


  Last clearing of "show modem" counters never
    0 incoming completes, 0 incoming failures
    0 outgoing completes, 0 outgoing failures
    0 failed dial attempts, 0 ring no answers, 1 busied outs
    0 no dial tones, 0 dial timeouts, 0 watchdog timeouts
    0 no carriers, 0 link failures, 0 resets, 0 recover oob
    0 protocol timeouts, 0 protocol errors, 0 lost events

Transmit Speed Counters:

Connection Speeds         75      300      600     1200     2400     4800
# of connections           0        0        0        0        0        0
Connection Speeds       7200     9600    12000    14400    16800    19200
# of connections           0        0        0        0        0        0
Connection Speeds      21600    24000    26400    28800    31200    32000
# of connections           0        0        0        1        0        0
Connection Speeds      33600    34000    36000    38000    40000    42000
# of connections           0        0        0        0        0        0
Connection Speeds      44000    46000    48000    50000    52000    54000
# of connections           0        0        0        0        0        0
Connection Speeds      56000
# of connections           0
```

Step 3 Enter line configuration mode and manually remove modem 1/3 from dial services by entering the **modem bad** command on line 4:

```
Router# configure terminal
Router(config)# line 4
Router(config-line)# modem bad
Router(config-line)# exit
Router(config)# exit
```

Step 4 Enter the **show modem** EXEC command or the **show modem** *slot*/*port* command to display the bad modem status.

Bad modems are marked with the letter B in the Mdm column of the **show modem** command display output.

```
Router# show modem

%SYS-5-CONFIG_I: Configured from console by consolem
                 Inc calls      Out calls   Busied   Failed  No       Succ
  Mdm   Usage    Succ   Fail    Succ   Fail   Out      Dial    Answer   Pct.
  1/0     0%       0      0       0      0       1        0       0      0%
  1/1     0%       0      0       0      0       3        0       0      0%
  1/2     0%       0      0       0      0       1        0       0      0%
B 1/3     0%       0      0       0      0       1        0       0      0%
  1/4     0%       0      0       0      0       1        0       0      0%
  1/5     0%       0      0       0      0       1        0       0      0%
  1/6     0%       0      0       0      0       1        0       0      0%
  1/7     0%       0      0       0      0       1        0       0      0%
  1/8     0%       0      0       0      0       1        0       0      0%
  1/9     0%       0      0       0      0       1        0       0      0%
  1/10    0%       0      0       0      0       1        0       0      0%
  1/11    0%       0      0       0      0       1        0       0      0%
  1/12    0%       0      0       0      0       1        0       0      0%
  1/13    0%       0      0       0      0       1        0       0      0%
  1/14    0%       0      0       0      0       1        0       0      0%
  1/15    0%       0      0       0      0       1        0       0      0%
  1/16    0%       0      0       0      0       1        0       0      0%
  1/17    0%       0      0       0      0       1        0       0      0%
  1/18    0%       0      0       0      0       0        0       0      0%
  1/19    0%       0      0       0      0       0        0       0      0%
  1/20    0%       0      0       0      0       0        0       0      0%
  1/21    0%       0      0       0      0       0        0       0      0%
  1/22    0%       0      0       0      0       0        0       0      0%
  1/23    0%       0      0       0      0       0        0       0      0%
```

Malfunctioning modems are also marked as Bad in the Status column of the **show modem** *slot*/*port* command display output, as the following example shows:

```
Router# show modem 1/3

Mdm  Typ    Status     Tx/Rx     G  Duration  TX  RX  RTS  CTS  DSR  DCD  DTR
1/3  V34    Bad     28800/28800  0  00:00:00           x    x    x    x    x

Modem 1/3, Microcom MNP10 V34 Modem (Managed), TTY4
Firmware (Boot) Rev: 1.0(23) (1.0(5))
Modem config: Incoming and Outgoing
Protocol: reliable/MNP, Compression: V42bis
Management port config: Status polling and AT session
Management port status: Status polling and AT session
TX signals: -15 dBm, RX signals: -17 dBm

  Last clearing of "show modem" counters never
    0 incoming completes, 0 incoming failures
    0 outgoing completes, 0 outgoing failures
```

```
    0 failed dial attempts, 0 ring no answers, 1 busied outs
    0 no dial tones, 0 dial timeouts, 0 watchdog timeouts
    0 no carriers, 0 link failures, 0 resets, 0 recover oob
    0 protocol timeouts, 0 protocol errors, 0 lost events

Transmit Speed Counters:

Connection Speeds         75      300      600     1200     2400     4800
# of connections           0        0        0        0        0        0
Connection Speeds       7200     9600    12000    14400    16800    19200
# of connections           0        0        0        0        0        0
Connection Speeds      21600    24000    26400    28800    31200    32000
# of connections           0        0        0        1        0        0
Connection Speeds      33600    34000    36000    38000    40000    42000
# of connections           0        0        0        0        0        0
Connection Speeds      44000    46000    48000    50000    52000    54000
# of connections           0        0        0        0        0        0
Connection Speeds      56000
# of connections           0
```

Clearing a Direct Connect Session on a Microcom Modem

The examples in this section are for Microcom modems.

The following example shows how to execute the **modem at-mode** command from a Telnet session:

```
Router# modem at-mode 1/1
```

The following example shows how to execute the **clear modem at-mode** command from a second Telnet session while the first Telnet session is connected to the modem:

```
Router# clear modem at-mode 1/1
clear "modem at-mode" for modem 1/1 [confirm] <press Return>
Router#
```

The following output is displayed in the first Telnet session after the modem is cleared by the second Telnet session:

```
Direct connect session cleared by vty0 (172.19.1.164)
```

Displaying Local Disconnect Reasons

To find out why a modem ended its connection or why a modem is not operating at peak performance, use the **show modem call-stats** [*slot*] EXEC command.

Disconnect reasons are described using four hexadecimal digits. The three lower-order digits can be used to identify the disconnect reason. The high-order digit generally indicates the type of disconnect reason or the time at which the disconnect occurred. For detailed information on the meaning of hexadecimal values for MICA modem disconnects, refer to the TAC Tech Notes document, *MICA Modem States and Disconnect Reasons*, at the following URL: http://www.cisco.com/warp/public/76/mica-states-drs.html

For detailed information on the meaning of hexadecimal values for NextPort modem disconnects, refer to the TAC Tech Notes document, *Interpreting NextPort Disconnect Reason Codes*, at the following URL: http://www.cisco.com/warp/public/471/np_disc_code.html .

Local disconnect reasons are listed across the top of the screen display (for example, wdogTimr, compress, retrain, inacTout, linkFail, moduFail, mnpProto, and lapmProt). In the body of the screen display, the number of times each modem disconnected is displayed (see the # column). For a particular disconnect reason, the % column indicates the percent that a modem was logged for the specified disconnect reason with respect to the entire modem pool for that given reason. For example, out of all the times the rmtLink error occurred on all the modems in the system, the rmtLink error occurred 10 percent of the time on modem 0/22.

Malfunctioning modems are detected by an unusually high number of disconnect counters for a particular disconnect reason. For example, if modem 1/0 had a high number of compression errors compared to the remaining modems in system, modem 1/0 would likely be the inoperable modem.

To reset the counters displayed by the **show modem call-stats** command, enter the **clear modem counters** command.

Note For a complete description of each error field displayed by the commands on this page, refer to the *Cisco IOS Dial Technologies Command Reference*. Remote disconnect reasons are not described by the **show modem** command output.

The following example displays output for the **show modem call-stats** command. Because of the screen size limitation of most terminal screen displays, not all possible disconnect reasons are displayed at one time. Only the top eight most frequently experienced disconnect reasons are displayed at one time.

```
Router# show modem call-stats

  dial-in/dial-out call statistics

       lostCarr  dtrDrop  rmtLink wdogTimr compress  retrain inacTout linkFail
  Mdm     #   %    #   %    #   %    #   %    #   %    #   %    #   %    #   %
* 0/0     6   2    2   3    1   0    0   0    0   0    0   0    0   0    0   0
* 0/1     5   2    2   3    2   1    0   0    0   0    0   0    0   0    0   0
  0/2     5   2    2   3    4   3    0   0    0   0    0   0    0   0    0   0
* 0/3     5   2    2   3    2   1    0   0    0   0    0   0    0   0    0   0
* 0/4     5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/5     5   2    2   3    2   1    0   0    0   0    0   0    0   0    0   0
* 0/6     4   1    2   3    2   1    0   0    0   0    0   0    0   0    0   0
* 0/7     4   1    2   3    4   3    0   0    0   0    0   0    0   0    0   0
* 0/8     6   2    1   1    3   2    0   0    0   0    0   0    0   0    0   0
* 0/9     5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/10    5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 0/11    5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
  0/12    5   2    2   3    2   1    0   0    0   0    0   0    0   0    0   0
* 0/13    5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/14    5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/15    5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/16    5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/17    5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 0/18    5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 0/19    5   2    1   1    3   2    0   0    0   0    0   0    0   0    0   0
* 0/20    5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/21    5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/22    5   2    1   1   11  10    0   0    0   0    0   0    0   0    0   0
* 0/23    5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 2/0     4   1    2   3    2   1    0   0    0   0    0   0    0   0    0   0
* 2/1     5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 2/2     5   2    2   3    0   0    0   0    0   0    0   0    0   0    0   0
* 2/3     5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 2/4     5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 2/5     5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 2/6     4   1    1   1    1   0    0   0    0   0    0   0    0   0    0   0
```

```
* 2/7      5    2     1    1     1    0     0    0     0    0     0    0     0    0     0    0
* 2/8      5    2     1    1     1    0     0    0     0    0     0    0     0    0     0    0
* 2/9      4    1     1    1     2    1     0    0     0    0     0    0     0    0     0    0
* 2/10     5    2     1    1     0    0     0    0     0    0     0    0     0    0     0    0
* 2/11     5    2     1    1     5    4     0    0     0    0     0    0     0    0     0    0
* 2/12     5    2     1    1     2    1     0    0     0    0     0    0     0    0     0    0
* 2/13     5    2     1    1     1    0     0    0     0    0     0    0     0    0     0    0
* 2/14     5    2     1    1     2    1     0    0     0    0     0    0     0    0     0    0
* 2/15     4    1     1    1     3    2     0    0     0    0     0    0     0    0     0    0
* 2/16     4    1     1    1     3    2     0    0     0    0     0    0     0    0     0    0
* 2/17     5    2     2    3     9    8     0    0     0    0     0    0     0    0     0    0
* 2/18     4    1     1    1     1    0     0    0     0    0     0    0     0    0     0    0
* 2/19     3    1     1    1     2    1     0    0     0    0     0    0     0    0     0    0
* 2/20     7    3     1    1     8    7     0    0     0    0     0    0     0    0     0    0
* 2/21     5    2     1    1     1    0     0    0     0    0     0    0     0    0     0    0
* 2/22     4    1     1    1     2    1     0    0     0    0     0    0     0    0     0    0
* 2/23     5    2     1    1     2    1     0    0     0    0     0    0     0    0     0    0
  Total   233        59       110         0         0         0         0         0
```

dial-out call statistics

```
          noCarr noDitone      busy     abort dialStrg autoLgon dialTout  rmtHgup
  Mdm      #   %    #   %     #   %     #   %     #   %     #   %     #   %     #   %
* 0/0      1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/1      0   0    0   0     0   0     0   0     0   0     0   0     0   0     0   0
  0/2      0   0    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/3      1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/4      1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/5      0   0    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/6      1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/7      5   5    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/8      0   0    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/9      1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/10     0   0    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/11     5   5    0   0     0   0     0   0     0   0     0   0     0   0     0   0
  0/12     0   0    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/13     0   0    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/14     1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/15     1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/16     2   2    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/17     4   4    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/18     5   5    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/19     1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/20     0   0    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/21     0   0    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/22     5   5    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 0/23     1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/0      2   2    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/1      3   3    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/2      0   0    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/3      0   0    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/4      0   0    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/5      1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/6      1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/7      4   4    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/8      7   8    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/9      4   1    1   1     2   1     0   0     0   0     0   0     0   0     0   0
* 2/10     1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/11     1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/12     1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/13     1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/14     4   4    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/15     1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
* 2/16     1   1    0   0     0   0     0   0     0   0     0   0     0   0     0   0
```

```
* 2/17     5    5     0    0     0    0     0    0     0    0     0    0     0    0     0    0
* 2/18     5    5     0    0     0    0     0    0     0    0     0    0     0    0     0    0
* 2/19     3    3     0    0     0    0     0    0     0    0     0    0     0    0     0    0
* 2/20     0    0     0    0     0    0     0    0     0    0     0    0     0    0     0    0
* 2/21     4    4     0    0     0    0     0    0     0    0     0    0     0    0     0    0
* 2/22     2    2     0    0     0    0     0    0     0    0     0    0     0    0     0    0
* 2/23     0    0     0    0     0    0     0    0     0    0     0    0     0    0     0    0
  Total    84         0         0         0         0         0         0         0         0
```

Removing Inoperable Modems

To manually remove inoperable modems from dialup services, use the following commands in line configuration mode:

	Command	Purpose
Step 1	`Router(config-line)# modem bad`	Removes and idles the modem from service and indicates it as suspected or proven to be inoperable.
Step 2	`Router(config-line)# modem hold-reset`	Resets and isolates the modem hardware for extensive troubleshooting.
Step 3	`Router(config-line)# modem shutdown`	Abruptly shuts down a modem from dial service.
Step 4	`Router(config-line)# modem recovery-time` *minutes*	Sets the maximum amount of time for which the call-switching module waits for a local modem to respond to a request before it is considered locked in a suspended state. The default is 5 minutes.

If you use the **modem bad** command to remove an idle modem from dial services and mark it as inoperable, the letter B is used to identify the modem as bad. The letter B appears in the Status column in the output of **show modem** *slot*/*port* command and in the far left column in the output of the **show modem** command. Use the **no modem bad** command to unmark a modem as B and restore it for dialup connection services. If the letter B appears next to a modem number, it means the modem was removed from service with the **modem shutdown** command.

Note Only idle modems can be marked "bad" by the **modem bad** command. If you want to mark a modem bad that is actively supporting a call, first enter the **modem shutdown** command, then enter the **modem bad** command.

Use the **modem hold-reset** command if a router is experiencing extreme modem behavior (for example, if the modem is uncontrollably dialing in to the network). This command prevents the modem from establishing software relationships such as those created by the **test modem back-to-back** command. The modem is unusable while the **modem hold-reset** command is configured. The **modem hold-reset** command also resets a modem that is frozen in a suspended state. Disable the suspended modem with the **modem hold-reset** command, and then restart hardware initialization with the **no modem hold-reset** command.

The following example disables a suspended modem and resets its hardware initialization:

```
Router# configure terminal
Router(config)# line 4
Router(config-line)# modem hold-reset
Router(config-line)# no modem hold-reset
```

The following example gracefully disables the modem associated with line 1 from dialing and answering calls. The modem is disabled only after all active calls on the modem are dropped.

```
Router# configure terminal
Router(config)# line 1
Router(config)# modem busyout
```

The following example abruptly shuts down the modem associated with line 2. All active calls on the modem are dropped immediately.

```
Router# configure terminal
Router(config)# line 2
Router(config)# modem shutdown
```

In the following example, the modem using TTY line 3 is actively supporting a call (as indicated by the asterisk). However, we want to mark the modem bad because it has poor connection performance. First, abruptly shut down the modem and drop the call with the **modem shutdown** command, and then enter the **modem bad** command to take the modem out of service.

```
Router# show modem

               Inc calls     Out calls     Busied   Failed  No       Succ
  Mdm  Usage   Succ   Fail   Succ   Fail   Out      Dial    Answer   Pct.
  1/0    37%     98      4      0      0       0        0       0     96%
  1/1    38%     98      2      0      0       0        0       0     98%
* 1/2    2%       3     99      0      0       0        0       0     1%
.
.
.

Router# configure terminal
Router(config)# line 3
Router(config)# modem shutdown
Router(config)# modem bad
Router(config)# exit

Router# show modem

               Inc calls     Out calls     Busied   Failed  No       Succ
  Mdm  Usage   Succ   Fail   Succ   Fail   Out      Dial    Answer   Pct.
  1/0    37%     98      4      0      0       0        0       0     96%
  1/1    38%     98      2      0      0       0        0       0     98%
B 1/2    2%       3     99      0      0       0        0       0     1%
```

For more information about modem recovery procedures, refer to TAC Tech Notes *Configuring MICA Modem Recovery* at http://www.cisco.com/warp/public/76/modem-recovery.html and *Configuring NextPort SPE Recovery* at http://www.cisco.com/warp/public/76/spe-recovery.html.

Busying Out a Modem Card

To busy out a modem card in a Cisco access server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line shelf/slot/port`	Specifies the line number, by specifying the shelf, slot, and port numbers; you must type in the slashes. This command also begins line configuration mode.
Step 2	`Router(config-line)# modem busyout`	Having specified the modem to be busied out with the **line** command, enter the **modem busyout** command to busy out the modem. The command disables the modem associated with line *shelf/slot/port* from dialing and answering calls.You need not specify a *shelf/slot/port* number again in this command.
Step 3	`Router(config-line)# modem shutdown`	Having specified the modem to be shut down with the **line** command, enter the **modem shutdown** command to shut down the modem, whether or not it has already been busied out. You need not specify a *shelf/slot/port* number again in this command because you have already done so with the **line** command.
Step 4	`Router(config-line)# exit`	Exits line configuration mode and returns to global configuration mode.
Step 5	`Router(config)# modem busyout-threshold number`	Specifies a threshold number using the **modem busyout-threshold** *number* command to balance the number of DS0s with the number of modem lines. For more information, refer to the *Cisco IOS Dial Technologies Command Reference*.
Step 6	`Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 7	`Router# show busyout`	From privileged EXEC mode, verifies that the line is busied out. If there are active calls, the software waits until the call terminates before the line is busied out.

The **modem busyout** command disables the modem associated with a specified line from dialing and answering calls. The **modem busyout** command can busy out and eventually terminate all 72 ports on the Cisco AS5800 modem card.

Monitoring Resources on Cisco High-End Access Servers

The following tasks enable you to monitor the network access server (NAS) health conditions at the DS0 level, PRI bearer channel level, and modem level. Performing these tasks will benefit network operation with improved visibility into the line status for the NAS for comprehensive health monitoring and notification capability, and improved troubleshooting and diagnostics for large-scale dial networks.

Perform the following tasks to monitor resource availability on the Cisco high-end access servers:

- Enabling DS0 Busyout Traps—DS0 busyout traps are generated when there is a request to busy out a DS0, when there is a request to take a DS0 out of busyout mode, or when busyout completes and the DS0 is out-of-service. DS0 busyout traps are generated at the DS0 level for both CAS and ISDN

configured lines. This feature is enabled and disabled through use of the CLI and MIBs. DS0 busyout traps are disabled by default and are supported on Cisco AS5300, Cisco AS5400, and Cisco AS5800 universal access servers.

- Enabling ISDN PRI Requested Channel Not Available Traps—ISDN PRI channel not available traps are generated when a requested DS0 channel is not available, or when there is no modem available to take the incoming call. This feature is available only for ISDN PRI interfaces. This feature is enabled and disabled through use of CLI for ISDN traps and the CISCO-ISDN-MIB. ISDN PRI channel not available traps are disabled by default and are supported on the Cisco AS5300, Cisco AS5400, and Cisco AS5800.
- Enabling Modem Health Traps—Modem health traps are generated when a modem port is bad, disabled, reflashed, or shut down, or when there is a request to busy out the modem. This feature is enabled and disabled through use of CLI and the CISCO-MODEM-MGMT-MIB. Modem health traps are disabled by default and are supported on the Cisco AS5300, Cisco AS5400, and Cisco AS5800.
- Enabling DS1 Loopback Traps—DS1 loopback traps are generated when a DS1 line goes into loopback mode. This feature is enabled and disabled by CLI and the CISCO-POP-MGMT-MIB. DS1 loopback traps are disabled by default and are supported on the Cisco AS5300 and Cisco AS5400 only.

The CISCO-POP-MGMT-MIB supplies the DS0 busyout traps and the DS1 loopback traps. The CISCO-MODEM-MGMT-MIB supplies additional modem health traps when the modem port becomes non-functional. The CISCO-ISDN-MIB supplies additional traps for ISDN PRI channel not available.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

See the sections “Verifying Enabled Traps” and “Troubleshooting the Traps” to verify and troubleshoot configuration. The section “NAS Health Monitoring Example” provides output of a configuration with the NAS health monitoring features enabled.

Enabling DS0 Busyout Traps

Before you enable DS0 busyout traps, the SNMP manager must already have been installed on your workstation, and the SNMP agent must be configured on the NAS by entering the **snmp-server community** and **snmp-server host** commands. Refer to the *Cisco IOS Configuration Fundamentals Configuration Guide* for more information on these commands.

To generate DS0 busyout traps, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server enable traps ds0-busyout`	Generates a trap when there is a request to busy out a DS0 or to indicate when busyout finishes.

Enabling ISDN PRI Requested Channel Not Available Traps

To generate ISDN PRI requested channel not available traps, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server enable traps isdn chan-not-avail`	Generates a trap when the NAS rejects an incoming call on an ISDN PRI interface because the channel is not available.

Enabling Modem Health Traps

To generate modem health traps, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server enable traps modem-health`	Generates a trap when a modem port is bad, disabled, or prepared for firmware download; when download fails; when placed in loopback mode for maintenance; or when there is a request to busy out the modem.

Enabling DS1 Loopback Traps

To generate DS1 loopback traps, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server enable traps ds1-loopback`	Generates a trap when the DS1 line goes into loopback mode.

Verifying Enabled Traps

To verify that the traps are enabled, use the **show run** command. The following output indicates that all the traps are enabled:

```
Router(config)# show run

snmp-server enable traps ds0-busyout
snmp-server enable traps isdn chan-not-avail
snmp-server enable traps modem-health
snmp-server enable traps ds1-loopback
```

Additionally, you can use the **show controllers** command with the **timeslots** keyword to display details about the channel state. This feature shows whether the DS0 channels of a particular controller are in idle, in-service, maintenance, or busyout state. This enhancement applies to both CAS and ISDN PRI interfaces and is supported on the Cisco AS5300 and Cisco AS5400 only.

Troubleshooting the Traps

To troubleshoot the traps, turn on the debug switch for SNMP packets by entering the following command in privileged EXEC mode:

```
Router# debug snmp packets
```

Check the resulting output to see that the SNMP trap information packet is being sent. The output will vary based on the kind of packet sent or received:

```
SNMP: Packet received via UDP from 10.5.4.1 on Ethernet0
SNMP: Get-next request, reqid 23584, errstat 0, erridx 0
sysUpTime = NULL TYPE/VALUE
 system.1 = NULL TYPE/VALUE
 system.6 = NULL TYPE/VALUE
SNMP: Response, reqid 23584, errstat 0, erridx 0
 sysUpTime.0 = 2217027
 system.1.0 = Cisco Internetwork Operating System Software
 system.6.0 =
SNMP: Packet sent via UDP to 10.5.4.1
```

You can also use trap monitoring and logging tools like snmptrapd, with debugging flags turned on, to monitor output.

NAS Health Monitoring Example

The following is sample configuration output showing all NAS health monitoring traps turned on:

```
Building configuration...

Current configuration:
! Last configuration change at 12:27:30 pacific Thu May 25 2000
version xx.x
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname router
!
aaa new-model
aaa authentication ppp default group radius
enable password <password>
!
spe 1/0 1/7
 firmware location system:/ucode/mica_port_firmware
spe 2/0 2/7
 firmware location system:/ucode/mica_port_firmware
!
resource-pool disable
!
clock timezone PDT -8
clock calendar-valid
no modem fast-answer
modem country mica usa
modem link-info poll time 60
modem buffer-size 300
ip subnet-zero
!
isdn switch-type primary-5ess
isdn voice-call-failure 0
!
```

```
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 1
 framing esf
 linecode b8zs
 ds0-group 0 timeslots 1-24 type e&m-fgb
 cas-custom 0
!
controller T1 2
 shutdown
 clock source line secondary 2
!
controller T1 3
 shutdown
 clock source line secondary 3
!
controller T1 4
 shutdown
 clock source line secondary 4
!
controller T1 5
 shutdown
 clock source line secondary 5
!
controller T1 6
 shutdown
 clock source line secondary 6
!
controller T1 7
 shutdown
 clock source line secondary 7
!
interface Loopback0
 ip address 10.5.4.1
!
interface Ethernet0
 no ip address
 shutdown
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface Serial2
 no ip address
 shutdown
!
interface Serial3
 no ip address
 shutdown
!
interface Serial0:23
 no ip address
 ip mroute-cache
 isdn switch-type primary-5ess
 isdn incoming-voice modem
```

```
 no cdp enable
!
interface FastEthernet0
 ip address 10.5.4.1
 duplex full
 speed auto
 no cdp enable
!
interface Group-Async1
 ip unnumbered FastEthernet0
 encapsulation ppp
 ip tcp header-compression passive
 no ip mroute-cache
 async mode interactive
 peer default ip address pool swattest
 no fair-queue
 ppp authentication chap
 ppp multilink
 group-range 1 192
!
interface Dialer1
 ip unnumbered FastEthernet0
 encapsulation ppp
 ip tcp header-compression passive
 dialer-group 1
 peer default ip address pool swattest
 pulse-time 0
 no cdp enable
!
ip local pool swattest 10.5.4.1
ip default-gateway 10.5.4.1
ip classless
!
dialer-list 1 protocol ip permit
snmp-server engineID local 00000009020000D058890CF0
snmp-server community public RO
snmp-server packetsize 2048
snmp-server enable traps ds0-busyout
snmp-server enable traps isdn chan-not-avail
snmp-server enable traps modem-health
snmp-server enable traps ds1-loopback
snmp-server host 10.5.4.1 public
!
radius-server host 10.5.4.1 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server key <password>
!
line con 0
 transport input none
line 1 192
 autoselect ppp
 modem InOut
 transport preferred none
 transport input all
 transport output none
line aux 0
line vty 0 4
end
```

Configuration Examples for Modem Management

This section provides the following examples:

- NextPort Modem Log Example
- Modem Performance Summary Example
- Modem AT-Mode Example
- Connection Speed Performance Verification Example

For additional information and examples about the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*.

NextPort Modem Log Example

The following is partial sample output for the Cisco AS5400 with the NextPort Distributed forwarding Card (DFC). This example shows the port history event log for slot 5, port 47:

```
Router# show port modem log 5/47

Port 5/47 Events Log
     Service type: DATA_FAX_MODEM
     Service mode: DATA_FAX_MODEM
     Session State: IDLE
  00:02:23: incoming called number: 35160
     Service type: DATA_FAX_MODEM
     Service mode: DATA_FAX_MODEM
     Session State: IDLE
     Service type: DATA_FAX_MODEM
     Service mode: DATA_FAX_MODEM
     Session State: ACTIVE
  00:02:23: Modem State event:
            State: Connect
  00:02:16: Modem State event:
            State: Link
  00:02:13: Modem State event:
            State: Train Up
  00:02:05: Modem State event:
            State: EC Negotiating
  00:02:05: Modem State event:
            State: Steady
  00:02:05: Modem Static event:
    Connect Protocol                          :   LAP-M
    Compression                               :   V.42bis
    Connected Standard                        :   V.34+
    TX,RX Symbol Rate                         :   3429, 3429
    TX,RX Carrier Frequency                   :   1959, 1959
    TX,RX Trellis Coding                      :   16/16
    Frequency Offset                          :   0  Hz
    Round Trip Delay                          :   0  msecs
    TX,RX Bit Rate                            :   33600, 33600
    Robbed Bit Signalling (RBS) pattern       :   0
    Digital Pad                               :   None
    Digital Pad Compensation                  :   None
    4 bytes of link info not formatted        :   0x00 0x00 0x00 0x00 0x00
  00:02:06:Modem Dynamic event:
    Sq Value                                  :   5
    Signal Noise Ratio                        :   40  dB
    Receive Level                             :   -12  dBm
    Phase Jitter Frequency                    :   0  Hz
```

```
    Phase Jitter Level                      :   2  degrees
    Far End Echo Level                      :   -90  dBm
    Phase Roll                              :   0  degrees
    Total Retrains                          :   0
    EC Retransmission Count                 :   0
    Characters transmitted, received        :   0, 0
    Characters received BAD                 :   0
    PPP/SLIP packets transmitted, received  :   0, 0
    PPP/SLIP packets received (BAD/ABORTED) :   0
    EC packets transmitted, received OK     :   0, 0
    EC packets (Received BAD/ABORTED)       :   0
```

Modem Performance Summary Example

You can display a high level summary of the performance of a modem with the **show modem summary** command:

```
Router# show modem summary

         Incoming calls       Outgoing calls      Busied   Failed   No    Succ
Usage  Succ   Fail  Avail   Succ   Fail  Avail    Out      Dial     Ans   Pct.
  14%  2489    123     15      0      0     15        0        3     3     95%
```

Modem AT-Mode Example

The following example shows that modem 1/1 has one open AT directly connected session:

```
Router# show modem at-mode

Active AT-MODE management sessions:
Modem    User's Terminal
1/1 0 cty 0
```

Connection Speed Performance Verification Example

Making sure that your modems are connecting at the correct connection speeds is an important aspect of managing modems. The **show modem connect-speeds** and **show modem** commands provide performance information that allow you to investigate possible inoperable or corrupt modems or T1/E1 lines. For example, suppose you have an access server that is fully populated with V.34 modems. If you notice that modem 1/0 is getting V.34 connections only 50 percent of the time, whereas all the other modems are getting V.34 connections 80 percent of the time, then modem 1/0 is probably malfunctioning. If you are reading low connection speeds across all the modems, you may have a faulty channelized T1 or ISDN PRI line connection.

To display connection speed information for all modems that are running in your system, use the **show modem connect-speeds** *max-speed* EXEC command. Because most terminal screens are not wide enough to display the entire range of connection speeds at one time (for example, 75 to 56,000 bps), the *max-speed* argument is used. This argument specifies the contents of a shifting baud-rate window, which provides you with a snapshot of the modem connection speeds for your system. Replace the *max-speed* argument with the maximum connect speed that you want to display. You can specify from 12,000 to 56,000 bps. If you are interested in viewing a snapshot of lower baud rates, specify a lower connection speed. If you are interested in displaying a snapshot of higher rates, specify a higher connection speed.

The following example displays connection speed information for modems running up to 33,600 bps:

```
Router# show modem connect-speeds 33600

  transmit connect speeds

   Mdm   14400  16800  19200  21600  24000  26400  28800  31200  33600 TotCnt
* 0/0       0      0      0      0      0      0      4      4      1      9
* 0/1       2      0      0      0      0      0      3      3      1      9
  0/2       2      0      0      0      0      1      2      4      1     10
* 0/3       0      0      0      1      0      0      3      4      1      9
* 0/4       1      0      0      0      0      2      2      1      1      7
* 0/5       0      0      0      0      0      0      4      4      1      9
* 0/6       0      0      0      0      0      1      3      3      1      8
* 0/7       0      0      0      2      0      0      4      3      1     10
* 0/8       2      0      0      0      0      0      3      4      1     10
* 0/9       0      0      0      0      0      0      4      3      0      7
* 0/10      1      0      0      0      0      1      3      2      1      8
* 0/11      0      0      0      0      0      0      4      3      1      8
  0/12      1      0      0      0      0      0      4      2      1      8
* 0/13      0      0      0      0      0      0      4      2      1      7
* 0/14      1      0      0      0      0      1      2      2      1      7
* 0/15      0      0      0      0      0      0      4      2      1      7
* 0/16      0      0      0      1      0      0      3      2      1      7
* 0/17      1      0      0      0      0      0      4      2      1      8
* 0/18      1      0      0      0      0      0      3      3      1      8
* 0/19      0      0      0      0      0      0      5      3      1      9
* 0/20      0      0      0      0      0      0      4      2      1      7
* 0/21      1      0      0      0      0      0      4      2      0      7
* 0/22      0      0      0      0      0      0      7      9      1     17
* 0/23      0      0      0      0      0      2      2      3      1      8
* 2/0       0      0      0      1      0      0      3      3      1      8
* 2/1       0      0      0      0      0      0      5      2      1      8
* 2/2       0      0      0      1      0      0      4      1      1      7
* 2/3       1      0      0      0      0      0      4      2      1      8
* 2/4       0      0      0      0      0      0      5      2      1      8
* 2/5       0      0      0      0      0      0      4      3      1      8
* 2/6       0      0      0      0      0      0      3      2      1      6
* 2/7       1      0      0      0      0      1      3      2      0      7
* 2/8       1      0      0      0      0      0      3      2      1      7
* 2/9       0      0      0      0      0      1      3      2      1      7
* 2/10      2      0      0      0      0      2      1      0      1      6
* 2/11      0      0      0      1      0      1      3      5      1     11
* 2/12      0      0      0      0      0      0      5      2      1      8
* 2/13      1      0      0      0      0      0      5      0      1      7
* 2/14      1      0      0      0      0      0      3      3      1      8
* 2/15      1      0      0      0      0      1      2      3      1      8
* 2/16      0      0      0      0      0      0      4      3      1      8
* 2/17      0      0      0      0      0      0      5     11      0     16
* 2/18      0      0      0      1      0      1      1      2      1      6
* 2/19      0      0      0      0      0      0      2      3      1      6
* 2/20      1      0      0      0      0      2      3      9      1     16
* 2/21      1      0      0      0      0      0      4      1      1      7
* 2/22      0      0      0      1      0      0      2      3      1      7
* 2/23      0      0      0      0      0      1      3      3      1      8
 Tot       23      0      0      9      0     18    165    141     44    400
 Tot %      5      0      0      2      0      4     41     35     11

  receive connect speeds

   Mdm   14400  16800  19200  21600  24000  26400  28800  31200  33600 TotCnt
* 0/0       0      0      0      0      0      4      1      3      1      9
* 0/1       2      0      0      0      0      3      1      2      1      9
  0/2       2      0      0      0      0      3      1      3      1     10
```

*	0/3	0	0	0	1	0	3	4	0	1	9
*	0/4	1	0	0	0	0	4	0	1	1	7
*	0/5	0	0	0	0	0	4	3	1	1	9
*	0/6	0	0	0	0	0	4	0	3	1	8
*	0/7	0	0	0	2	0	4	1	2	1	10
*	0/8	2	0	0	0	0	3	0	5	0	10
*	0/9	0	0	0	0	0	4	2	0	1	7
*	0/10	1	0	0	0	0	4	0	2	1	8
*	0/11	0	0	0	0	0	4	0	3	1	8
	0/12	1	0	0	0	0	2	2	2	1	8
*	0/13	0	0	0	0	0	4	1	1	1	7
*	0/14	1	0	0	0	0	2	3	0	1	7
*	0/15	0	0	0	0	0	4	1	1	1	7
*	0/16	0	0	0	1	0	3	2	0	1	7
*	0/17	1	0	0	0	0	4	1	1	1	8
*	0/18	1	0	0	0	0	3	2	1	1	8
*	0/19	0	0	0	0	0	5	1	2	1	9
*	0/20	0	0	0	0	0	4	0	3	0	7
*	0/21	1	0	0	0	0	4	0	1	1	7
*	0/22	0	0	0	0	0	6	6	4	1	17
*	0/23	0	0	0	0	0	4	2	1	1	8
*	2/0	0	0	0	1	0	3	1	2	1	8
*	2/1	0	0	0	0	0	3	3	1	1	8
*	2/2	0	0	0	1	0	4	0	1	1	7
*	2/3	1	0	0	0	0	3	2	1	1	8
*	2/4	0	0	0	0	0	4	2	1	1	8
*	2/5	0	0	0	0	0	4	1	2	1	8
*	2/6	0	0	0	0	0	3	0	3	0	6
*	2/7	1	0	0	0	1	2	2	0	1	7
*	2/8	1	0	0	0	0	3	0	2	1	7
*	2/9	0	0	0	0	0	4	1	1	1	7
*	2/10	2	0	0	0	0	3	0	0	1	6
*	2/11	0	0	0	1	0	3	1	5	1	11
*	2/12	0	0	0	0	0	4	3	0	1	8
*	2/13	1	0	0	0	0	2	3	0	1	7
*	2/14	1	0	0	0	0	3	2	1	1	8
*	2/15	1	0	0	0	0	3	0	3	1	8
*	2/16	0	0	0	0	0	4	0	4	0	8
*	2/17	0	0	0	0	0	5	2	8	1	16
*	2/18	0	0	1	0	0	2	1	1	1	6
*	2/19	0	0	0	0	0	2	2	1	1	6
*	2/20	1	0	0	0	0	4	2	8	1	16
*	2/21	1	0	0	0	0	4	0	1	1	7
*	2/22	0	0	1	0	0	2	0	3	1	7
*	2/23	0	0	0	0	0	4	2	1	1	8
	Tot	23	0	2	7	1	167	64	92	44	400
	Tot %	5	0	0	1	0	41	16	23	11	

Configuring and Managing Cisco Access Servers and Dial Shelves

This chapter describes configuration and monitoring tasks for the Cisco AS5800 and AS5400 access servers, including dial shelves and dial shelf controllers on the Cisco AS5800 access servers in the following main sections:

- Cisco AS5800 Dial Shelf Architecture and DSIP Overview
- How to Configure Dial Shelves
- Port Management Services on Cisco Access Servers
- Upgrading and Configuring SPE Firmware

For further information and configuration examples for the Cisco AS5400, refer to the *Cisco AS5400 Universal Access Server Software Configuration Guide*.

For further information and configuration examples for the Cisco AS5800, refer to the *Cisco AS5800 Universal Access Server Operations, Administration, Maintenance, and Provisioning Guide*.

For more information on the Cisco access servers, go to the Cisco Connection Documentation site on Cisco.com, or use the Cisco Documentation CD-ROM.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

For a complete description of the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

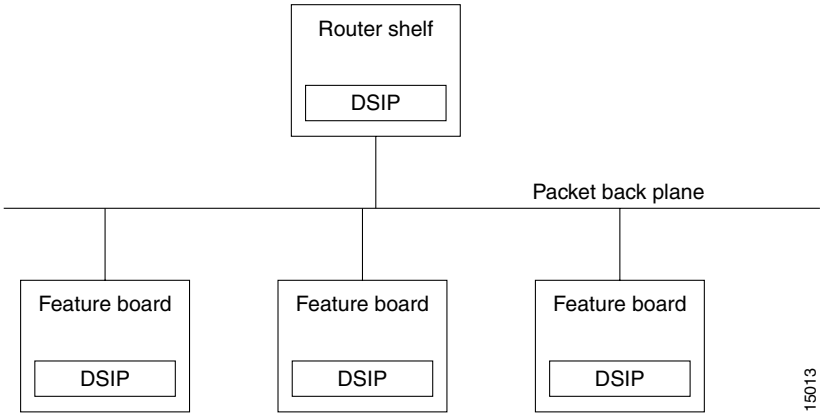
Cisco AS5800 Dial Shelf Architecture and DSIP Overview

The Cisco AS5800 is a rack-mounted system consisting of a router shelf and a dial shelf. The dial shelf contains feature and controller cards (trunk cards), modem cards, and dial shelf controller (DSC) cards.

Note For more information about split dial shelf configuration, refer to the hardware installation guides that accompanied your Cisco AS5800 Universal Access Server and the *Cisco AS5800 Universal Access Server Software Installation and Configuration Guide*.

The Dial Shelf Interconnect Protocol (DSIP) is used for communication between router shelf and dial shelf on an AS5800. Figure 22 diagrams the components of the architecture. The router shelf is the host for DSIP commands, which can be run remotely on the feature boards of the dial shelf using the command, **execute-on**. DSIP communicates over the packet backplane via the dial shelf interconnect (DSI) cable.

Figure 22 DSIP Architecture in the Cisco AS5800



Split Dial Shelves Feature

The split dial shelves feature provides for doubling the throughput of the Cisco AS5800 access server by splitting the dial shelf slots between two router shelves, each router connected to one Dial Shelf Controller (DSC), two of which must be installed in the system. Each router shelf is configured to control a certain set from the range of the dial shelf slots. Each router shelf will operate as though any other slots in the dial shelf contained no cards, even if there is a card in them, because they are controlled by the other router shelf. Thus the configuration on each router shelf would affect only the "owned" slots.

Each router shelf should own modem cards and trunk cards. Calls received on a trunk card belonging to one router shelf cannot be serviced by a modem card belonging to the other router shelf. Each router shelf operates like a single Cisco AS5800 access server system, as if some slots are unavailable.

Refer to the section "Configuring Dial Shelf Split Mode" for more information about configuring split dial shelves.

How to Configure Dial Shelves

To configure and maintain dial shelves, perform the tasks in the following sections:

- Configuring the Shelf ID
- Configuring Redundant DSC Cards
- Synchronizing to the System Clocks
- Configuring Dial Shelf Split Mode
- Executing Commands Remotely
- Verifying DSC Configuration

- Monitoring and Maintaining the DSCs
- Troubleshooting DSIP

Configuring the Shelf ID

The Cisco AS5800 consists of a router shelf and a dial shelf. To distinguish the slot/port number on the Cisco AS5800, you must specify the shelf number. The default shelf number is 0 for the router shelf and 1 for the dial shelf.

Caution You must reload the Cisco AS5800 for the new shelf number to take effect. Because the shelf number is part of the interface names when you reload, all NVRAM interface configuration information is lost.

Normally you do not need to change the shelf IDs; however, if you do, we recommend that you change the shelf number when you initially access the setup facility. For information on the setup facility, refer to the *Cisco AS5800 Universal Access Server Software Installation and Configuration Guide*.

If you are booting the router shelf from the network (netbooting), you can change the shelf numbers using the **shelf-id** command.

To configure the dial shelf, you save and verify the configuration in EXEC mode, and enter **shelf-id** commands in global configuration mode, as indicated in the following steps:

	Command	Purpose
Step 1	`Router# copy startup-configure tftp`	Saves your current configuration. Changing the shelf number removes all interface configuration information when you reload the Cisco AS5800.
Step 2	`Router# configure terminal`	Begins global configuration mode.
Step 3	`Router(config)# shelf-id` *number* `router-shelf`	Specifies the router shelf ID.
Step 4	`Router(config)# shelf-id` *number* `dial-shelf`	Specifies the dial shelf ID.
Step 5	`Router(config)# exit`	Exits global configuration mode.
Step 6	`Router# copy running-config startup-config`	Saves your configuration. This step is optional.
Step 7	`Router# show version`	Verifies that the correct shelf number will be changed after the next reload.
Step 8	`Router# reload components all`	Instructs the DSC (or DSCs in a redundant configuration) be reloaded at the same time as a reload on the router shelf. Type “yes” to the “save config” prompt. Configure one interface so that its router shelf has connectivity to the server with the configuration.
Step 9	`Router# copy tftp startup-config`	Because changing the shelf number removes all interface configuration information when you reload the Cisco AS5800, edit the configuration file saved in step 1 and download it.

If you are booting the router shelf from Flash memory, use the following commands beginning in EXEC mode:

	Command	Purpose

Step 1	`Router# copy running-config tftp`
	or
	`Router# copy startup-config tftp`
		Saves your current (latest) configuration to a server.

Step 2	`Router# configure terminal`	Begins global configuration mode.
Step 3	`Router(config)# shelf-id number router-shelf`	Configures the router shelf ID.
Step 4	`Router(config)# shelf-id number dial-shelf`	Configures the dial shelf ID.
Step 5	`Router(config)# exit`	Exits global configuration mode.
Step 6	`Router> copy running-config startup-config`	Saves your configuration. This step is optional. If this step is skipped, type "No" at the "save configuration" prompt.
Step 7	`Router> show version`	Allows verification that the correct shelf number will be changed after the next reload. Edit the configuration file saved in Step 1.
Step 8	`Router> copy tftp startup-config`	Copies the edited configuration to NVRAM on the Cisco AS5800.
Step 9	`Router# reload components all`	Instructs the DSC (or DSCs in a redundant configuration) to be reloaded at the same time as a reload on the router shelf.

Configuring Redundant DSC Cards

The Redundant Dial Shelf Controller feature consists of two DSC cards on a Cisco AS5800 dial shelf. The DSC cards provide clock and power control to the dial shelf cards. Each DSC card provides the following:

- Master clock for the dial shelf
- Fast Ethernet link to the router shelf
- Environmental monitoring of the feature boards
- Bootstrap images on start-up for the feature boards

The Redundant Dial Shelf Controller feature is automatically enabled when two DSC cards are installed. DSC redundancy is supported with Cisco AS5800 software at the Dial Shelf Interconnect Protocol (DSIP) level.

This feature enables a Cisco AS5800 dial shelf to use dual DSCs for full redundancy. A redundant configuration allows for one DSC to act as backup to the active card, should the active card fail. This increases system availability by preventing loss of service. The redundant DSC functionality is robust under high loads and through DSC or software crashes and reloads. The redundant DSC functionality is driven by the following events:

- User actions
- Control messages
- Timeouts

- Detection of component failures
- Error and warning messages

DSC redundancy provides maximum system availability by preventing loss of service if one of the DSCs fails. There is no load sharing between the Broadband Inter-Carrier Interfaces (BICI). One BIC is used as a backup, carrying only control traffic, such as keepalives, until there is a switchover.

Before starting this configuration task:

- Your Cisco AS5800 router shelf and dial shelf must be fully installed, with two DSC cards installed on the dial shelf.
- Your Cisco AS5800 access server must be running Cisco IOS Release 12.1(2)T.
- The external DSC clocking port must be configured identically on both router shelves and must be physically connected to both DSCs. This assures that if a DSC card needs replacing or if the backup DSC card becomes primary, clocking remains stable.

Synchronizing to the System Clocks

The time-division multiplexing (TDM) bus in the backplane on the dial shelf must be synchronized to the T1/E1 clocks on the trunk cards. The Dial Shelf Controller (DSC) card on the daily shelf provides hardware logic to accept multiple clock sources as input and use one of them as the primary source to generate a stable, PPL synchronized output clock. The input clock can be any of the following sources:

- Trunk port in slots 0 through 5—up to 12 can be selected (2 per slot)
- An external T1 or E1 clock source fed directly through a connector on the DSC card
- A free-running clock from an oscillator in the clocking hardware on the DSC card

For dual (redundant) DSC cards, the external DSC clocking port should be configured so that the clock signal fed into both DSCs is identical.

To configure the external clocks, use the following commands from the router shelf login beginning in global configuration mode. One external clock is configured as the primary clock source, and the other is configured as the backup clock source.

	Command	Purpose
Step 1	`Router(config)# dial-tdm-clock priority value`	Configures the trunk card clock priority. Priority range is a value between 1 and 50.
Step 2	`Router(config)# dial-tdm-clock priority X {trunk-slot Y port Z} external {t1 | e1} [120-ohm]`	Selects the T1/E1 trunk slot and port that is providing the clocking source. T1/E1 selection is based on the incoming signal. Select the impedance. The default impedance is 75-ohm.

Step 3	`Router(config)# dial-tdm-clock priority value external t1`
	or
	`Router(config)# dial-tdm-clock priority value external e1`
		Configures the T1/E1 external clock on the dial shelf controller front panel. T1/E1 selection is based on the signal coming in. Priority range is a value between 1 and 50.

Step 4	`Router(config)# Ctrl-Z` `Router#`	Verifies your command registers when you press the return key. Enter **Ctrl-Z** to return to privileged EXEC mode.
Step 5	`Router# copy running-config startup-config`	Saves your changes.

Verifying External Clock Configuration

To verify that the primary clock is running, enter the **show dial-shelf clocks** privileged EXEC command:

```
Router# show dial-shelf 12 clocks

Slot 12:
System primary is 1/2/0 of priority 202
TDM Bus Master Clock Generator State = NORMAL
Backup clocks:
Source  Slot    Port    Priority      Status      State
-------------------------------------------------------
Trunk   2       1       208            Good        Default
Slot    Type    11 10  9  8  7  6  5  4  3  2  1  0
2       T1       G  G  G  G  G  G  G  G  G  G  G  G
```

For more information on configuring external clocks, refer to the Cisco document *Managing Dial Shelves.*

Configuring Dial Shelf Split Mode

This section describes the procedure required to transition a router from normal mode to split mode and to change the set of slots a router owns while it is in split mode. Since the process of switching the ownership of a slot from one router to the other is potentially disruptive (when a feature board is restarted, all calls through that card are lost), a router shelf cannot take over a slot until ownership is relinquished by the router that currently claims ownership, either by reconfiguring the router or disconnecting that router or its associated DSC.

The dial shelf is split by dividing the ownership of the feature boards between the two router shelves. You must configure the division of the dial shelf slots between the two router shelves so that each router controls an appropriate mix of trunk and modem cards. Each router shelf controls its set of feature boards as if those were the only boards present. There is no interaction between feature boards owned by one router and feature boards owned by the other router.

Split mode is entered when the **dial-shelf split slots** command is parsed on the router shelf. This can occur when the router is starting up and parsing the stored configuration, or when the command is entered when the router is already up. Upon parsing the **dial-shelf split slots** command, the router frees any resources associated with cards in the slots that it no longer owns, as specified by exclusion of slot numbers from the *slot-numbers* argument. The router should be in the same state as if the card had been removed from the slot; all calls through that card will be terminated. The configured router then informs its connected DSC that it is in split mode, and which slots it claims to own.

In split mode, a router shelf by default takes half of the 2048 available TDM timeslots. The TDM split mode is configured using the **dial-shelf split backplane-ds0** command. (The **dial-shelf split slot** command must be defined for the **dial-shelf split backplane-ds0** command to be active.) If the **dial-shelf split slots** command is entered when the total number of calls using timeslots exceeds the number that would normally be available to the router in split mode, the command is rejected. This should occur only when a change to split mode is attempted, in which the dial shelf has more than 896 calls in progress (more than half of the 1,792 available timeslots). Otherwise, a transition from normal mode to split mode can be made without disturbing the cards in the slots that remain owned, and calls going through those cards will stay up.

To configure a router for split dial shelf operation, perform the following steps:

Step 1 Ensure that both DSCs and both router shelves are running the same Cisco IOS image.

Note Having the same version of Cisco IOS running on both DSCs and both router shelves is not mandatory; however, it is a good idea. There is no automatic checking that the versions are the same.

Step 2 Schedule a time when the Cisco AS5800 can be taken out of service without unnecessarily terminating calls in progress. The entire procedure for transitioning from normal mode to split mode should require approximately one hour if all the hardware is already installed.

Step 3 Busy out all feature boards and wait for your customers to log off.

Step 4 Reconfigure the existing router shelf to operate in split mode.

Step 5 Enter the **dial-shelf split slots** command, specifying the slot numbers that are to be owned by the existing router shelf.

Step 6 Configure the new router shelf to operate in split mode on other feature boards.

Step 7 Enter the **dial-shelf split slots** command, specifying the slot numbers that are to be owned by the new router shelf. Do not specify any of the slot numbers that you specified in Step 6. The range of valid slot numbers is 0 through 11.

To perform this step, enter the following command in global configuration mode:

Command	Purpose
`Router(config)# dial-shelf split slots` *slot-numbers*	Enter list of slot numbers, for example: `dial-shelf split slots 0 1 2 6 7 8` In this example, the other router shelf could be configured to own the other slots: 3 4 5 9 10 11. **Normal mode**: This command changes the router shelf to split mode with ownership of the slots listed. In case of conflicting slot assignments, the command is rejected and a warning message is issued. Issue a **show dial-shelf split slots** command to the other router shelf to display its list of owned dial shelf slots. Online insertion and removal (OIR) events on all slots are detected by both DSCs and added to the list of feature boards physically present in the dial shelf; however, OIR event processing is done only for assigned slots. **Split mode**: This command adds the dial shelf slots listed to the router shelf's list of owned dial shelf slots.

Step 8 Install the second DSC, if it has not already been installed.

Step 9 Connect the DSIP cable from the second DSC to the new router shelf.

Step 10 Ensure that split mode is operating properly.

Enter the **show dial-shelf** command for each router. This command has been extended so that the response indicates that the router shelf is running in split mode and which slots the router shelf owns. The status of any cards in any owned slots is shown, just as they are in the present **show dial-shelf** command. When in split mode, the output will be extended as in the following example:

```
System is in split dial shelf mode.
Slots owned: 0 2 3 4 5 6 (connected to DSC in slot 13)
Slot    Board     CPU       DRAM          I/O Memory   State   Elapsed
         Type     Util    Total (free)   Total (free)          Time
 0        CE1    0%/0%  21341728( 87%)  8388608( 45%)  Up      00:11:37
 2        CE1    0%/0%  21341728( 87%)  8388608( 45%)  Up      00:11:37
 4 Modem(HMM)  20%/20%   6661664( 47%)  6291456( 33%)  Up      00:11:37
 5 Modem(DMM)    0%/0%   6661664( 31%)  6291456( 32%)  Up      00:11:37
 6 Modem(DMM)    0%/0%   6661664( 31%)  6291456( 32%)  Up      00:11:37
13        DSC    0%/0%  20451808( 91%)  8388608( 66%)  Up      00:16:31
Dial shelf set for auto boot
```

Step 11 Enable all feature boards to accept calls once again.

Changing Slot Sets

You can change the sets of slots owned by the two router shelves while they are in split mode by first removing slots from the set owned by one router, and then adding them to the slot set of the other router. The changed slot set information is sent to the respective DSCs, and the DSCs determine which slots have been removed and which added from the new slot set information. It should be clear that moving a slot in this manner will disconnect all calls that were going through the card in that slot.

To perform this task, enter the following commands as needed:

Command	Purpose
`Router (config)# dial-shelf split slots remove` *slot-numbers*	Removes the dial shelf slots listed from the router shelf's list of owned dial shelf slots. The effect of multiple commands is cumulative.
`Router(config)# dial-shelf split slots` *slot-numbers*	Adds the dial shelf slots listed to the router shelf's list of owned dial shelf slots.

When a Slot Is Removed

The router shelf that is losing the slot frees any resources and clears any state associated with the card in the slot it is relinquishing. The DSC reconfigures its hub to ignore traffic from that slot, and if there is a card in the slot, it will be reset. This ensures that the card frees up any TDM resource it might be using and allows it to restart under control of the router shelf that is subsequently configured to own the slot.

When a Slot Is Added

If there are no configuration conflicts, and there is a card present in the added slot, a dial-shelf OIR insertion event is sent to the router shelf, which processes the event the same as it always does. The card in the added slot is reset by the DSC to ensure a clean state, and the card downloads its image from the router shelf that now owns it.

If the other router shelf and the other DSC claim ownership of the same slot, the command adding the slot should be rejected. However, should a configuration conflict exist, error messages are sent to both routers and the card is not reset until one of the other router shelves and its DSC stop claiming ownership of the slot. Normally, this will not happen until you issue a **dial-shelf split slots remove** command surrendering the ownership claim on the slot by one of the routers.

Leaving Split Mode

Split mode is exited when the dial shelf configuration is changed by a **no dial-shelf split slots** command. When the split dial shelf line is removed, the router shelf will start using all of the TDM timeslots. Feature boards that were not owned in split mode and that are not owned by the other router will be reset. Cards in slots that are owned by the other router will be reset, but only after the other DSC has been removed or is no longer claiming the slots. The split dial shelf configuration should not be removed while the second router shelf is still connected to the dial shelf.

When a router configured in split mode fails, all calls associated with the failed router are lost. Users cannot connect back in until the failed router recovers and is available to accept new incoming calls; however, the other split mode router shelf will continue to operate normally.

Troubleshooting Split Dial Shelves

The system will behave as configured as soon as the configuration is changed. The exception is when there is a misconfiguration, such as when one router is configured in split mode and the other router is configured in normal mode, or when both routers are configured in split mode and both claim ownership of the same slots.

Problems can arise if one of the two routers connected to a dial shelf is not configured in split mode, or if both are configured in split mode and both claim ownership of the same slots. If the state of the second router is known when the **dial-shelf split slots** command is entered and the command would result in a conflict, the command is rejected.

If a conflict in slot ownership does arise, both routers will receive warning messages until the conflict is resolved. Any card in a slot which is claimed by both routers remains under the control of the router that claimed it first, until you can resolve the conflict by correcting the configuration of one or both routers.

It should be noted that there can also be slots that are not owned by either router (orphan slots). Cards in orphan slots cannot boot up until one of the two routers claims ownership of the slot because neither DSC will download bootstrap images to cards in unowned orphan slots.

Managing a Split Dial Shelf

If you are installing split dial shelf systems, a system controller is available that provides a single system view of multiple point of presences (POPs). The system controller for the Cisco AS5800 Universal Access Server includes the Cisco 3640 router running Cisco IOS software. The system controller can be installed at a remote facility so that you can access multiple systems through a console port or Web interface.

There are no new MIBs or MIB variables required for the split dial shelf configuration. A split dial shelf appears to Simple Network Management Protocol (SNMP) management applications as two separate Cisco AS5800 systems. One console to manage the whole system is not supported—you must have a console session per router shelf (two console sessions) to configure each split of the Cisco AS5800. The system controller must manage a split dial shelf configuration as two separate Cisco AS5800 systems.

The normal mode configuration of the Cisco AS5800 requires the dial shelf and router shelf IDs to be different. In a split system, four unique shelf IDs are desirable, one for each router shelf and one for each of the slot sets; however, a split system will function satisfactorily if the router shelf IDs are the same. If a system controller is used to manage a split dial shelf configuration, the two routers must have distinct shelf IDs, just as they must when each router has its own dial shelf.

You can download software configurations to any Cisco AS5800 using SNMP or a Telnet connection. The system controller also provides performance monitoring and accounting data collection and logging.

In addition to the system controller, a network management system with a graphical user interface (GUI) runs on a UNIX SPARC station and includes a database management system, polling engine, trap management, and map integration.

To manage a split dial shelf, enter the following commands in EXEC mode as needed:

Command	Purpose
`Router# show dial-shelf split`	Displays the slots assigned to each of the router shelves and the corresponding feature boards in 'orphan' slots (slots not currently assigned to either router).
`Router# show dial-shelf`	Displays information about the dial shelf, including clocking information.
`Router# show context`	Displays information about the dial shelf, including clocking information, but works only for owned slots. Use **show context all** to display all the information available about any slot. This is intended to cover the case where ownership of a feature board is moved from one router shelf to the other after a crash.

Executing Commands Remotely

Although not recommended, it is possible to connect directly to the system console interface in the DSC to execute dial shelf configuration commands. All commands necessary for dial shelf configuration, and **show**, and **debug** command tasks can be executed remotely from the router console. A special command, **execute-on**, is provided for this purpose. This command enables a special set of EXEC mode commands to be executed on the router or the dial shelf. This command is a convenience that avoids connecting the console to the DSC. For a list of commands you can execute using **execute-on**, refer to the command description in the *Cisco IOS Dial Technologies Command Reference*.

To enter a command that you wish to execute on a specific card installed in the dial shelf while logged onto the router shelf console, use the following commands in privileged EXEC mode as needed:

Command	Purpose
`Router# execute-on slot` *slot command*	Executes a command from the router shelf on a specific slot in the dial shelf.
`Router# execute-on all` *command*	Executes a command from the router shelf on all cards in the dial shelf.

Verifying DSC Configuration

To verify that you have started the redundant DSC feature, enter the **show redundancy** privileged EXEC command:

```
Router# show redundancy

DSC in slot 12:

Hub is in 'active' state.
Clock is in 'active' state.

DSC in slot 13:

Hub is in 'backup' state.
Clock is in 'backup' state.

Router#
```

Monitoring and Maintaining the DSCs

To monitor and maintain the DSC cards, use the following commands in privileged EXEC mode, as needed:

Command	Purpose
Router# **hw-module** *shelf/slot* {**start**|**stop**}	Stops the target DSC remotely from the router console. Restart the DSC if it has been stopped.
Router# **show redundancy** [**history**]	Displays the current or history status for redundant DSC.
Router# **debug redundancy** {**all**|**ui**|**clk**|**hub**}	Use this debug command if you need to collect events for troubleshooting, selecting the appropriate required key word.
Router# **show debugging**	Lists the debug commands that are turned on, including those for redundant DSC.

Troubleshooting DSIP

There are a number of show commands available to aid in troubleshooting dial shelves. Use the following EXEC mode commands to monitor DSI and DSIP activity as needed:

Command	Purpose
Router# **clear dsip tracing**	Clears tracing statistics for the DSIP.
Router# **show dsip**	Displays all information about the DSIP.
Router# **show dsip clients**	Displays information about DSIP clients.
Router# **show dsip nodes**	Displays information about the processors running the DSIP.
Router# **show dsip ports**	Displays information about local and remote ports.
Router# **show dsip queue**	Displays the number of messages in the retransmit queue waiting for acknowledgment.
Router# **show dsip tracing**	Displays DSIP tracing buffer information.

Command	Purpose
`Router# show dsip transport`	Displays information about the DSIP transport statistics for the control/data and IPC packets and registered addresses.
`Router# show dsip version`	Displays DSIP version information.

The privileged EXEC mode **show dsi** command can also be used to troubleshoot, as it displays the status of the DSI adapter, which is used to physically connect the router shelf and the dial shelf to enable DSIP communications.

The following is an example troubleshooting scenario:

Problem: The router shelf boots, but there is no communication between the router and dial shelves.

Step 1 Run the **show dsip transport** command.

Step 2 Check the "DSIP registered addresses" column. If there are zero entries there, there is some problem with the Dial Shelf Interconnect (DSI). Check if the DSI is installed in the router shelf.

Step 3 If there is only one entry and it is our own local address, then first sanity check the physical layer. Make sure that there is a physical connection between the RS and DS. If everything is fine from cabling point of view, go to step 3.

Step 4 Check the DSI health by issuing the **show dsi** command. This gives a consolidated output of DSI controller and interface. Check for any errors like runts, giants, throttles and other usual FE interface errors.

Diagnosis: If an entry for a particular dial shelf slot is not found among the registered addresses, but most of other card entries are present, the problem is most likely with that dial shelf slot. The DSI hardware on that feature board is probably bad.

Port Management Services on Cisco Access Servers

Port Management Services on the Cisco AS5400 Access Server

Port service management on the Cisco AS5400 access server implements service using the NextPort dial feature card (DFC). The NextPort DFC is a hardware card that processes digital service port technology for the Cisco AS5400 access server. A port is defined as an endpoint on a DFC card through which multiservice tones and data flow. The ports on the NextPort DFC support both modem and digital services. Ports can be addressed-aggregated at the slot level of the NextPort module, the Service Processing Element (SPE) level within the NextPort module, and the individual port level. Cisco IOS Release 12.1(3)T or higher is required for the NextPort DFC.

Instead of the traditional line-modem one-to-one correspondence, lines are mapped to an SPE that resides on the Cisco AS5400 NextPort DFC. Each SPE provides modem services for six ports. Busyout and shutdown can be configured at the SPE or port level. The NextPort DFC introduces the slot and SPE software hierarchy. On the Cisco AS5400, the hierarchy designation is *slot*/*SPE*.

The NextPort DFC slot is defined as a value between 1 and 7. Slot 0 is reserved for the motherboard. Each NextPort DFC provides 18 SPEs. The SPE value ranges from 0 to 17. Since each SPE has six ports, the NextPort DFC has a total of 108 ports. The port value ranges from 0 to 107.

The NextPort DFC performs the following functions:

- Converts pulse code modulation (PCM) bitstreams to digital packet data.
- Forwards converted and packetized data to the main processor, which examines the data and forwards it to the backhaul egress interface.
- Supports all modem standards (such as V.34 and V.42*bis*) and features, including dial-in and dial-out.

Port Management Services on the Cisco AS5800 Access Server

Port service management on the Cisco AS5800 access server implements service on the Universal Port Card (UPC). A universal port carries a single channel at the speed of digital signal level 0 (DS0), or the equivalent of 64-kbps on a T1 facility.

Network traffic can be a modem, voice, or fax connection. The 324 port UPC uses NextPort hardware and firmware to provide universal ports for the Cisco AS5800 access server. These ports are grouped into 54 service processing elements (SPEs). Each SPE supports six universal ports. To find the total number of ports supported by a UPC, multiply the 54 SPEs by the six ports supported on each SPE. The total number of universal ports supported by a single UPC is 324. Configuration, management, and troubleshooting of universal ports can be done at the UPC, SPE, and port level. Each UPC also has a SDRAM card with a minimum of a 128 MB of memory.

The Cisco AS5800 access server can be equipped with a maximum of seven UPCs with upgradable firmware. The UPC supports data traffic, and depending on the software and platform is universal port capable. Each UPC plugs directly into the dial shelf backplane and does not need any external connections. Each UPC has three LEDs, which indicate card status.

The Cisco AS5800 access server is capable of terminating up to 2,048 incoming modem connections (slightly more than an OC3) when equipped with seven UPCs and three CT3 trunk cards. A split shelf configuration with a second router shelf and second dial shelf controller are required to achieve full capacity. A single router with a standard configuration supports up to 1,344 port connections. Cisco IOS Release 12.1(3)T or higher is required for the UPC. Unless your system shipped with UPCs installed, you must upgrade the Cisco IOS image on the dial shelf and router shelf or shelves.

Instead of the traditional line-modem one-to-one correspondence, lines are mapped to an SPE that resides on the Cisco AS5800 access server UPC. Each SPE provides modem services for six ports. Busyout and shutdown can be configured at the SPE or port level. The UPC introduces the shelf, slot, and SPE software hierarchy. On the Cisco AS5800 access server, the hierarchy designation is *shelf*/*slot*/*SPE*.

A UPC can be installed in slots numbered 2 to 11 on the dial shelf backplane. If installed in slots 0 or 1, the UPC automatically powers down. Slots 0 and 1 only accept trunk cards; they do not accept mixes of cards. We recommend that you install mixes of T3 and T1 cards, or E1 trunk cards in slots 2 to 5. You can use double-density modem cards, UPCs, and VoIP cards simultaneously. Trunk cards can operate in slots 0 to 5 and are required for call termination.

The UPC performs the following functions:

- Converts pulse code modulation (PCM) bitstreams to digital packet data.
- Forwards converted and packetized data to the dial shelf main processor, which examines the data and forwards it to the router shelf. From the router shelf, the data is routed to the external network.

- Supports all modem standards (such as V.34 and V.42*bis*) and features, including dial-in and dial-out.
- Supports online insertion and removal (OIR), a feature that allows you to remove and replace UPCs while the system is operating. A UPC can be removed without disrupting the operation of other cards and their associated calls. If a UPC is removed while the system is operating, connections or current calls on that card are dropped. Calls being handled by other cards are not affected.

Note All six ports on an SPE run the same firmware.

Upgrading and Configuring SPE Firmware

SPE firmware is automatically downloaded in both the Cisco AS5400 and AS5800 access servers.

AS5400 Access Server

SPE firmware is automatically downloaded to a NextPort DFC from the Cisco AS5400 when you boot the system for the first time, or when you insert a NextPort DFC while the system is operating. When you insert DFCs while the system is operating, the Cisco IOS image recognizes the cards and downloads the required firmware to the cards.

The SPE firmware image is bundled with the access server Cisco IOS image. The SPE firmware image uses an *autodetect* mechanism, which enables the NextPort DFC to service multiple call types. An SPE detects the call type and automatically configures itself for that operation. For further information on upgrading SPE firmware from the Cisco IOS image, refer to the section "Configuring SPEs to Use an Upgraded Firmware File."

The firmware is upgradeable independent of Cisco IOS upgrades, and different firmware versions can be configured to run on SPEs in the same NextPort DFC. You can download firmware from the Cisco System Cisco.com File Transfer Protocol (FTP) server.

AS5800 Access Server

SPE firmware is automatically downloaded to an AS5800 UPC from the router shelf Cisco IOS image when you boot the system for the first time or when you insert a UPC while the system is operating. The Cisco IOS image recognizes the card and the dial shelf downloads the required portware to the cards. Cisco IOS Release 12.1(3)T or higher is required for the UPC.

The SPE firmware image (also known as *portware*) is bundled with the Cisco IOS UPC image. The SPE firmware image uses an *autodetect* mechanism, which enables the UPC to service multiple call types. An SPE detects the call type and automatically configures itself for that operation. For further information on upgrading SPE firmware from the Cisco IOS image, refer to the section "Configuring SPEs to Use an Upgraded Firmware File."

The firmware is upgradable independent of Cisco IOS upgrades, and different firmware versions can be configured to run on SPEs in the same UPC. You can download firmware from the Cisco.com File Transfer Protocol (FTP) server.

Firmware Upgrade Task List

Upgrading SPE firmware from the Cisco.com FTP server is done in two steps:

- Downloading SPE Firmware from the Cisco.com FTP Server to a Local TFTP Server
- Copying the SPE Firmware File from the Local TFTP Server to the SPEs

Firmware Configuration Task List

To complete firmware configuration once you have downloaded the SPE firmware, perform the tasks in the following sections:

- Specifying a Country Name
- Configuring Dial Split Shelves (AS5800 Only)
- Configuring SPEs to Use an Upgraded Firmware File
- Disabling SPEs
- Rebooting SPEs
- Configuring Lines
- Configuring Ports
- Verifying SPE Line and Port Configuration
- Configuring SPE Performance Statistics
- Clearing Log Events
- Troubleshooting SPEs
- Monitoring SPE Performance Statistics

Note The following procedure can be used for either a Cisco AS5400 or AS5800 access server.

Downloading SPE Firmware from the Cisco.com FTP Server to a Local TFTP Server

Note You must be a registered Cisco user to log in to the Cisco Software Center.

You can download software from the Cisco Systems Cisco.com FTP server using an Internet browser or using an FTP application. Both procedures are described.

Using an Internet Browser

Step 1 Launch an Internet browser.

Step 2 Bring up the Cisco Software Center home page at the following URL (this is subject to change without notice):

http://www.cisco.com/kobayashi/sw-center/

Step 3 Click **Access Software** (under Cisco Software Products) to open the Access Software window.

Step 4 Click **Cisco AS5400 Series** or **Cisco AS5800 Series** software.

Step 5 Click the SPE firmware you want and download it to your workstation or PC. For example, to download SPE firmware for the universal access server, click **Download Universal Images**.

Step 6 Click the SPE firmware file you want to download, and then follow the remaining download instructions. If you are downloading the SPE firmware file to a PC, make sure that you download the file to the c:/tftpboot directory; otherwise, the download process does not work.

Step 7 When the SPE firmware is downloaded to your workstation, transfer the file to a Trivial File Transfer Protocol (TFTP) server in your LAN using a terminal emulation software application.

Step 8 When the SPE firmware is downloaded to your workstation, transfer the file to a TFTP server somewhere in your LAN using a terminal emulation software application.

Using an FTP Application

Note The directory path leading to the SPE firmware files on cco.cisco.com is subject to change without notice. If you cannot access the files using an FTP application, try the Cisco Systems URL http://www.cisco.com/cgi-bin/ibld/all.pl?i=support&c=3.

Step 1 Log in to the Cisco.com FTP server called cco.cisco.com:

```
terminal> ftp cco.cisco.com
Connected to cio-sys.cisco.com.
```

Step 2 Enter your registered username and password (for example, **harry** and **letmein**):

```
Name (cco.cisco.com:harry): harry
331 Password required for harry.
Password: letmein
230-###############################################################
230-#  Welcome to the Cisco Systems CCO FTP server.
230-#  This server has a number of restrictions.  If you are not familiar
230-#  with these, please first get and read the /README or /README.TXT file.
230-#  http://www.cisco.com/acs/info/cioesd.html for more info.
230-###############################################################
```

Step 3 Specify the directory path that holds the SPE firmware you want to download. For example, the directory path for the Cisco AS5400 SPE firmware is /cisco/access/5400:

```
ftp> cd /cisco/access/5400
250-Please read the file README
250-  it was last modified on Tue May 27 10:07:38 1997 - 48 days ago
250-Please read the file README.txt
250-  it was last modified on Tue May 27 10:07:38 1997 - 48 days ago
250 CWD command successful.
```

Step 4 Enter the **ls** command to view the contents of the directory:

```
ftp> ls
227 Entering Passive Mode (192,31,7,130,218,128)
150 Opening ASCII mode data connection for /bin/ls.
total 2688
drwxr-s--T   2 ftpadmin ftpcio    512 Jun 30 18:11 .
drwxr-sr-t  19 ftpadmin ftpcio    512 Jun 23 10:26 ..
lrwxrwxrwx  1  root     3          10 Aug   6 1996     README ->README.txt
-rw-rw-r--   1 root     ftpcio   2304 May 27 10:07 README.txt
-r--r--r-- 1 ftpadmin ftpint 377112 Jul 10 18:08 np-spe-upw-10.0.1.2.bin
-r--r--r-- 1 ftpadmin ftpint 635 Jul 10 18:08 SPE-firmware.10.1.30.readme
```

Step 5 Specify a binary image transfer:

```
ftp> binary
200 Type set to I.
```

Step 6 Copy the SPE firmware files from the access server to your local environment with the **get** command.

Step 7 Quit your terminal session:

```
ftp> quit
Goodbye.
```

Step 8 Enter the **ls -al** command to verify that you successfully transferred the files to your local directory:

```
server% ls -al
total 596
-r--r--r-- 1 280208 Jul 10 18:08 np-spe-upw-10.0.1.2.bin
server% pwd
/auto/tftpboot
```

Step 9 Transfer these files to a local TFTP or remote copy protocol (RCP) server that your access server or router can access.

Copying the SPE Firmware File from the Local TFTP Server to the SPEs

The procedure for copying the SPE firmware file from your local TFTP server to the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs is a two-step process. First, transfer the SPE firmware to the access server's Flash memory. Then, configure the SPEs to use the upgrade firmware. The upgrade occurs automatically, either as you leave configuration mode, or as specified in the configuration.

These two steps are performed only once. After you copy the SPE firmware file into Flash memory for the first time, you should not have to perform these steps again.

Note Because the SPE firmware is configurable for individual SPEs or ranges of SPEs, the Cisco IOS software automatically copies the SPE firmware to each SPE each time the access server restarts.

To transfer SPE Firmware to Flash memory, perform the following task to download the Universal SPE firmware to Flash memory:

Step 1 Check the image in the access server Flash memory:

```
Router# show flash
System flash directory:
File  Length   Name/status
  1   4530624  c5400-js-mx
 [498776 bytes used, 16278440 available, 16777216 total]
16384K bytes of processor board System flash (Read/Write)
```

Step 2 Enter the **copy tftp flash** command to download the code file from the TFTP server into the access server Flash memory. You are prompted for the download destination and the remote host name.

```
Router# copy tftp flash
```

Step 3 Enter the **show flash** command to verify that the file has been copied into the access server Flash memory:

```
Router# show flash
```

Specifying a Country Name

To set the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs to be operational for call set up, you must specify the country name. To specify the country name, use the following command in global configuration mode:

Command	Purpose
`Router(config)# spe country` *country name*	Specifies the country to set the UPC or DFC parameters (including country code and encoding). If you do not specify a country, the interface uses the default. If the access server is configured with T1 interfaces, the default is **usa**. If the access server is configured with E1 interfaces, the default is **e1-default**. Use the **no** form of this command to set the country code to the default of the domestic country. **Note** All sessions in all UPCs or DFCs in all slots must be in the idle state for this command to execute.

Configuring Dial Split Shelves (AS5800 Only)

The Cisco AS5800 access server requires a split dial shelf configuration using two router shelves to achieve the maximum capacity of 2048 port connections using the seven UPCs and three T3 + 1 T1 trunks. A new configuration command is available to define the split point:

dial-shelf split backplane-ds0 *option*

The options for this command come in pairs, and vary according to the desired configuration. You will need to log in to each router shelf and separately configure the routers for the intended load. In most circumstances it is recommended that the predefined options are selected. These options are designed to be matched pairs as seen below.

	Router Shelf 1			Router Shelf 2			Total
Option Pair	Option	Maximum Calls	Unused T1	Option	Maximum Calls	Unused T1	
1	**2ct3cas**	1344		**1ct3cas**	672		2016
2	**part2ct1ct3cas**	1152	4	**part1ct1ct3cas**	888	3	2040
3	**2ct3isdn**	1288		**part1ct1ct3isdn_b**	644	7	1932
4	**part2ct1ct3isdn**	1150	2	**part1ct1ct3isdn**	897	1	2047
5[1]	**3ce1**	960		**3ce1**	960		1920
6	Default (no option entered)	1/2 of current input		Default (no option entered)	1/2 of current input		
7	**no dial-shelf backplane-ds0**	1024		**no dial-shelf backplane-ds0**	1024		2048

1. This option is used to revert to the default for an environment using 6 E1 lines.

The **dial-shelf split slot 0 3 4 5** command must be defined for the **dial-shelf split backplane-ds0** *option* command to be active. You may also select the **user defined** option to define your own split.

Even if your system is already using a split dial shelf configuration, configuring one router shelf to handle two T3 trunks and the other router to handle the third trunk requires you to take the entire access server out of service. Busyout all connections before attempting to reconfigure. The configuration must be changed to setup one pool of TDM resources that can be used by either DMM cards or UPCs, and a second pool of two streams that contains TDM resources that can only be used by UPCs.

You may have more trunk capacity than 2048 calls. It is your decision how to provision the trunks so the backplane capacity is not exceeded. If more calls come in than backplane DS0 capacity for that half of the split, the call will be rejected and an error message printed for each call. This cannot be detected while a new configuration is being built because the router cannot tell which T1 trunks are provisioned and which are not. The user may want some trunks in hot standby.

The DMM, HMM, and VoIP cards can only use 1792 DS0 of the available 2048 backplane DS0. The UPC and trunk cards can use the full 2048 backplane DS0. The **show tdm splitbackplane** command will show the resources in two groups, the first 1792 accessible to all cards, and the remaining 256 accessible only to UPC and trunk cards.

For more information about split dial shelf configuration, refer to the *Cisco AS5800 Universal Access Server Split Dial Shelf Installation and Configuration Guide* and the hardware installation guides that accompanied your Cisco AS5800 Universal Access Server.

Configuring SPEs to Use an Upgraded Firmware File

To configure the SPEs to use the upgraded firmware file, use the following commands beginning in privileged EXEC mode to display the firmware version number:

	Command	Purpose
Step 1	`Router# show spe version`	Displays SPE firmware versions to obtain the On-Flash firmware filename.
Step 2	`Router# configure terminal`	Enters global configuration mode.

Step 3	AS5400:
	`Router(config)# spe slot/spe`
	or
	`Router(config)# spe slot/spe slot/spe`
	
	AS5800:
	`Router(config)# spe shelf/slot/spe`
	or
	`Router(config)# spe shelf/slot/spe shelf/slot/spe`
		Enters the SPE configuration mode. You can choose to configure a range of SPEs by specifying the first and last SPE in the range.

Step 4	`Router(config-spe)# firmware upgrade {busyout | download-maintenance | reboot}`	Specifies the upgrade method. Three methods of upgrade are available. The **busyout** keyword waits until all calls are terminated on an SPE before upgrading the SPE to the designated firmware. The **download-maintenance** keyword upgrades the firmware during the download maintenance time. The **reboot** keyword requests the access server to upgrade firmware at the next reboot.

	Command	Purpose
Step 5	`Router(config-spe)# firmware location filename`	Specifies the SPE firmware file in Flash memory to use for the selected SPEs. Allows you to upgrade firmware for SPEs after the new SPE firmware image is copied to your Flash memory. Enter the **no firmware location** command to revert back to the default Cisco IOS bundled SPE firmware.
Step 6	`Router(config-spe)# exit`	Exits SPE configuration mode.
Step 7	`Router# exit`	Exits global configuration mode.
Step 8	`Router# copy running-config startup-config`	Saves your changes.

Note The **copy ios-bundled** command is not necessary with UPCs or NextPort DFCs. By default, the version of SPE firmware bundled with the Cisco IOS software release transfers to all SPEs not specifically configured for a different SPE firmware file.

Disabling SPEs

To disable specific SPEs in the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs, use the following commands starting in global configuration mode:

	Command	Purpose

Step 1	**Cisco AS5400 Series Routers**
	`Router(config)# spe slot/spe`
	or
	`Router(config)# spe slot/spe slot/spe`
	Cisco AS5800 Series Routers
	`Router(config)# spe shelf/slot/spe`
	or
	`Router(config)# spe shelf/slot/spe shelf/slot/spe`
		Enters SPE configuration mode. You can also configure SPEs specifying the first and last SPE in a range.

	Command	Purpose
Step 2	`Router(config-spe)# busyout`	Gracefully disables an SPE by waiting for all the active services on the specified SPE to terminate. You can perform auto-diagnostic tests and firmware upgrades when you put the SPEs in the Busy out state. Active ports on the specified SPE will change the state of the specified range of SPEs to the BusyoutPending state. The state changes from BusyoutPending to Busiedout when all calls end. Use the **show spe** command to see the state of the range of SPEs. Use the **no** form of this command to re-enable the SPEs.
Step 3	`Router(config-spe)# shutdown`	Clears active calls on all ports on the SPE. Calls can no longer be placed on the SPE because the SPE state is changed to Busiedout. Use the **no** form of this command to re-enable the ports on the SPE.

Rebooting SPEs

To reboot specified SPEs, use the following command in privileged EXEC mode:

Command	Purpose

Cisco AS5400 Series Routers
`Router# clear spe slot/spe`
 	
Cisco AS5800 Series Routers
`Router# clear spe shelf/slot/spe`
	Allows manual recovery of a port that is frozen in a suspended state. Reboots SPEs that are in suspended or Bad state. Downloads configured firmware to the specified SPE or range of SPEs and power-on self test (POST) is executed.
	
	Note Depending on the problem, sometimes downloading the SPE firmware may not help recover a bad port or an SPE.
	
	This command can be executed regardless of the state of SPEs. All active ports running on the SPE are prematurely terminated, and messages are logged into the appropriate log.

Configuring Lines

To configure the lines to dial in to your network, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	**Cisco AS5400 Series Routers**
	`Router(config)# line slot/port slot/port`
	Cisco AS5800 Series Routers
	`Router(config)# line shelf/slot/port shelf/slot/port`
		Enters the line configuration mode. You can specify a range of slot and port numbers to configure.
		On the Cisco AS5400 access server, the NextPort DFC slot is defined as a value between 1 and 7. Slot 0 is reserved for the motherboard. Each NextPort DFC provides 18 SPEs. The SPE value ranges from 0 to 17. Since each SPE has six ports, the NextPort DFC has a total of 108 ports. The port value ranges from 0 to 107. To configure 108 ports on slot 3, you would enter **line 3/00 3/107**. If you wish to configure 324 ports on slots 3-5, you would enter **line 3/00 5/107**.
		On the Cisco AS5800 access server, the UPC slot is defined as a value between 2 and 11. Each UPC provides 54 SPEs. The SPE value ranges from 0 to 53. Because each SPE has six ports, the UPC has a total of 324 ports. The port value ranges from 0 to 323. To configure 324 ports on slot 3, you would enter **line 1/3/00 1/3/323**. If you want to configure 972 ports on slots 3-5, you would enter **line 1/3/00 1/5/323**.

Step 2	`Router(config-line)# transport input all`	Allows all protocols when connecting to the line.
Step 3	`Router(config-line)# autoselect ppp`	Enables remote IP users running a PPP application to dial in, bypass the EXEC facility, and connect directly to the network.
Step 4	`Router(config-line)# modem inout`	Enables incoming and outgoing calls.
Step 5	`Router(config-line)# modem autoconfigure type name`	Configures the attached modem using the entry for name.

Configuring Ports

This section describes how to configure Cisco AS5800 UPC or Cisco AS5400 NextPort DFC ports. You need to be in port configuration mode to configure these ports. The port configuration mode allows you to shut down or put individual ports or ranges of ports in busyout mode. To configure Cisco AS5800 UPC or Cisco AS5400 NextPort DFC ports, perform the following tasks beginning in global configuration mode:

	Command	Purpose

Step 1	**Cisco AS5400 Series Routers**
	`Router(config)# port slot/spe`
	or
	`Router(config)# port slot/spe slot/spe`
	Cisco AS5800 Series Routers
	`Router(config)# port shelf/slot/spe`
	or
	`Router(config)# port shelf/slot/spe shelf/slot/spe`
		Enters port configuration mode. You can choose to configure a single port or range of ports.

Step 2	`Router(config-port)# busyout`	(Optional) Gracefully disables a port by waiting for the active services on the specified port to terminate. Use the **no** form of this command to re-enable the ports. Maintenance activities, such as testing, can still be performed while the port is in busyout mode. **Note** When a port is in busyout mode, the state of the SPE is changed to the consolidated states of all the underlying ports on that SPE.
Step 3	`Router(config-port)# shutdown`	(Optional) Clears active calls on the port. No more calls can be placed on the port in the shutdown mode. Use the **no** form of this command to re-enable the ports. **Note** When a port is in shutdown mode, the state of the SPE is changed to the consolidated states of all the underlying ports on that SPE.
Step 4	`Router(config-port)# exit`	Exits port configuration mode.

Verifying SPE Line and Port Configuration

To verify your SPE line configuration, enter the **show spe** command to display a summary for all the lines and ports:

Step 1 Enter the **show spe** command to display a summary for all the lines and ports:

```
Router# show spe
```

Step 2 Enter the **show line** command to display a summary for a single line.

AS5400

```
Router# show line 1/1
```

AS5800

```
Router# show line 1/2/10
```

Note If you are having trouble, make sure that you have turned on the protocols for If you are having trouble, make sure that you have turned on the protocols for connecting to the lines (**transport input all**) and that your access server is configured for incoming and outgoing calls (**modem inout**).

Configuring SPE Performance Statistics

Depending on the configuration, call record is displayed on the console, or the syslog, or on both. The log contains raw data in binary form, which must be viewed using the **show** commands listed in the section "Monitoring SPE Performance Statistics." You can configure some aspects of history events by using one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# spe call-record modem` *max-userid*	Requests the access server to generate a modem call record after a call is terminated. To disable this function, use the **no** form of this command.
`Router(config)# spe log-size` *number*	Sets the maximum size of the history event queue log entry for each port. The default is 50 events per port.

Clearing Log Events

To clear some or all of the log events relating to the SPEs as needed, use the following privileged EXEC mode commands:

Command	Purpose
`Router# clear spe log`	Clears all event entries in the slot history event log.
`Router# clear spe counters`	Clears statistical counters for all types of services for the specified SPE, a specified range of SPEs, or all SPEs. If you do not specify the range of SPEs or an SPE, the statistics for all SPEs are cleared.
`Router# clear port log`	Clears all event entries in the port level history event log. You cannot remove individual service events from the port log.

Troubleshooting SPEs

This section provides troubleshooting information for your SPEs regardless of service type mode.

Note SPE ports that pass the diagnostic test are marked as Pass, Fail, and Unkn. Ports that fail the diagnostic test are marked as Bad. These ports cannot be used for call connections. Depending on how many ports are installed, the diagnostic tests may take from 5 to 10 minutes to complete.

- Enter the **port modem startup-test** command to perform diagnostic testing for all modems during the system's initial startup or rebooting process. To disable the test, enter the **no port modem startup-test** command.
- Enter the **port modem autotest** command to perform diagnostic testing for all ports during the system's initial startup or rebooting process.To disable the test, enter the **no port modem autotest** command.

  You may additionally configure the following options:

  – Enter the **port modem autotest minimum** *ports* command to define the minimum number of free ports available for autotest to begin.

  – Enter the **port modem autotest time** *hh*:*mm interval* command to enable autotesting time and interval.

  – Enter the **port modem autotest error** *threshold* command to define the maximum number of errors detected for autotest to begin.
- Enter the **show port modem test** command to displays results of the SPE port startup test and SPE port auto-test.

When an SPE port is tested as Bad, you may perform additional testing by conducting a series of internal back-to-back connections and data transfers between two SPE ports. All port test connections occur inside the access server. For example, if mobile users cannot dial into port 2/5 (which is the sixth port on the NextPort DFC in the second chassis slot), attempt a back-to-back test with port 2/5 and a known-functioning port such as port 2/6.

- Enter the **test port modem back-to-back** *slot/port slot/port* command to perform internal back-to-back port tests between two ports sending test packets of the specified size.

Note You might need to enable this command on several different combinations of ports to determine which one is not functioning properly. A pair of operable ports successfully connects and completes transmitting data in both directions. An operable port and an inoperable port do not successfully connect with each other.

A sample back-to-back test might look like the following:

```
Router# test port modem back-to-back 2/10 3/20
Repetitions (of 10-byte packets) [1]:
*Mar  02 12:13:51.743:%PM_MODEM_MAINT-5-B2BCONNECT:Modems (2/10) and (3/20) connected
in back-to-back test:CONNECT33600/V34/LAP
*Mar  02 12:13:52.783:%PM_MODEM_MAINT-5-B2BMODEMS:Modems (3/20) and (2/10) completed
back-to-back test:success/packets = 2/2
```

Tips You may reboot the port that has problems using the **clear spe** EXEC command.

- Enter the **spe recovery** {**port-action** {**disable** | **recover** | **none**} | **port-threshold** *num-failures*} command to perform automatic recovery (removal from service and reloading of SPE firmware) of ports on an SPE at any available time.

  An SPE port failing to connect for a certain number of consecutive times indicates that a problem exists in a specific part or the whole of SPE firmware. Such SPEs have to be recovered by downloading firmware. Any port failing to connect *num-failures* times is moved to a state based on the **port-action** value, where you can choose to disable (mark the port as Bad) or recover the port when the SPE is in the idle state and has no active calls. The default for *num-failures* is 30 consecutive call failures.

Tips You may also schedule recovery using the **spe download maintenance** command.

- Enter the **spe download maintenance time** *hh*:*mm* | **stop-time** *hh*:*mm* | **max-spes** *number* | **window** *time-period* | **expired-window** {**drop-call** | **reschedule**} command to perform a scheduled recovery of SPEs.

  The download maintenance activity starts at the set start **time** and steps through all the SPEs that need recovery and the SPEs that need a firmware upgrade and starts maintenance on the maximum number of set SPEs for maintenance. The system waits for the **window** delay time for all the ports on the SPE to become inactive before moving the SPE to the Idle state. Immediately after the SPE moves to Idle state, the system starts to download firmware. If the ports are still in use by the end of **window** delay time, depending upon the **expired-window** setting, connections on the SPE ports are shutdown and the firmware is downloaded by choosing the **drop-call** option, or the firmware download is rescheduled to the next download maintenance time by choosing the **reschedule** option. This process continues until the number of SPEs under maintenance is below **max-spes,** or until **stop-time** (if set), or until all SPEs marked for recovery or upgrade have had their firmware reloaded.

Monitoring SPE Performance Statistics

This section documents various SPE performance statistics for the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs:

- SPE Events and Firmware Statistics
- Port Statistics
- Digital SPE Statistics
- SPE Modem Statistics

SPE Events and Firmware Statistics

To view SPE events and firmware statistics for the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs, use one or more of the following commands in privileged EXEC mode:

Command	Purpose

Cisco AS5400 series routers
`Router# show spe slot/spe`
Cisco AS5800 series routers
`Router# show spe shelf/slot/spe`
	Displays the SPE status for the specified range of SPEs.

`Router# show spe log [reverse | slot]`	Displays the SPE system log.
`Router# show spe version`	Lists all SPEs and the SPE firmware files used. **Note** This list helps you decide if you need to update your SPE firmware files.

Port Statistics

To view port statistics for the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs, use the following commands in privileged EXEC mode as needed:

Command	Purpose

Cisco AS5400 series routers
`Router# show port config {slot | slot/port}`
Cisco AS5800 series routers
`Router# show port config {slot | shelf/slot/port}`
	Displays the configuration information for specified ports or the specified port range. The port should have an active session associated at the time the command is executed.

Cisco AS5400 series routers `Router# show port digital log [reverse slot/port] [slot | slot/port]`	Displays the digital data event log.

Command	Purpose
Cisco AS5400 series routers `Router# show port modem log [reverse slot/port] [slot | slot/port]` **Cisco AS5800 series routers** Router# **show port modem log** [**reverse** *shelf/slot/port*] [*shelf/slot* | *shelf/slot/port*]	Displays the port history event log.

Cisco AS5400 series routers
`Router# show port modem test [slot | slot/port]`
Cisco AS5800 series routers
`Router# show port modem test [shelf/slot | shelf/slot/port]`
	Displays the test log for the specified SPE port range or all the SPE ports.

Cisco AS5400 series routers
`Router# show port operational-status [slot | slot/port]`
Cisco AS5800 series routers
`Router# show port operational-status [shelf/slot | shelf/slot/port]`
	Displays the operational status of the specified ports or the specified port range. The port should have an active session associated at the time the command is executed.

Digital SPE Statistics

To view digital SPE statistics for the Cisco AS5400 NextPort DFCs, use one or more of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show spe digital [slot | slot/spe]`	Displays history statistics of all digital SPEs.
`Router# show spe digital active [slot | slot/spe]`	Displays active digital statistics of a specified SPE, the specified range of SPEs, or all the SPEs.
`Router# show spe digital csr [summary | slot | slot/spe]`	Displays the digital call success rate statistics for a specific SPE, a range of SPEs, or all the SPEs.
`Router# show spe digital disconnect-reason [summary | slot | slot/spe]`	Displays the digital disconnect reasons for the specified SPE or range of SPEs. The disconnect reasons are displayed with Class boundaries.
`Router# show spe digital summary [slot | slot/spe]`	Displays digital history statistics of all SPEs, a specified SPE, or the specified range of SPEs for all service types.

SPE Modem Statistics

To view SPE modem statistics for the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs, use one or more of the following commands in privileged EXEC mode:

Command	Purpose

Cisco AS5400 series routers
`Router# show spe modem active {slot | slot/spe}`
Cisco AS5800 series router:
`Router# show spe modem active {shelf/slot | shelf/slot/spe}`
	Displays the active statistics of a specified SPE, a specified range of SPEs, or all the SPEs serving modem traffic.

Cisco AS5400 series routers
`Router# show spe modem csr {summary | slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem csr {summary | shelf/slot | shelf/slot/spe}`
	Displays the call success rate statistics for a specific SPE, range of SPEs, or all the SPEs.

Cisco AS5400 series routers
`Router# show spe modem disconnect-reason {summary | slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem disconnect-reason {summary | shelf/slot | shelf/slot/spe}`
	Displays the disconnect reasons for the specified SPE or range of SPEs. The disconnect reasons are displayed with Class boundaries.

Cisco AS5400 series routers
`Router# show spe modem high speed {summary | slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem high speed {summary | shelf/slot | shelf/slot/spe}`
	Shows the connect-speeds negotiated within each high speed modulation or codecs for a specific range of SPEs or all the SPEs.

Cisco AS5400 series routers
`Router# show spe modem low speed {summary | slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem low speed {summary | shelf/slot | shelf/slot/spe}`
	Shows the connect-speeds negotiated within each low speed modulation or codecs for a specific range of SPEs or all the SPEs.

Cisco AS5400 series routers
`Router# show spe modem high standard {summary | slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem high standard {summary | shelf/slot | shelf/slot/spe}`
	Displays the total number of connections within each low modulation or codec for a specific range of SPEs.

Command	Purpose

Cisco AS5400 series routers
`Router# show spe modem low standard {summary | slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem low standard {summary | shelf/slot | shelf/slot/spe}`
	Displays the total number of connections within each high modulation or codec for a specific range of SPEs.

Cisco AS5400 series routers
`Router# show spe modem summary {slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem summary {shelf/slot | shelf/slot/spe}`
	Displays the history statistics of all SPEs, specified SPE or the specified range of SPEs.

Configuring and Managing External Modems

This chapter describes how to configure externally connected modems. These tasks are presented in the following main sections:

- External Modems on Low-End Access Servers
- Automatically Configuring an External Modem
- Manually Configuring an External Modem
- Supporting Dial-In Modems
- Testing the Modem Connection
- Managing Telnet Sessions
- Modem Troubleshooting Tips
- Checking Other Modem Settings

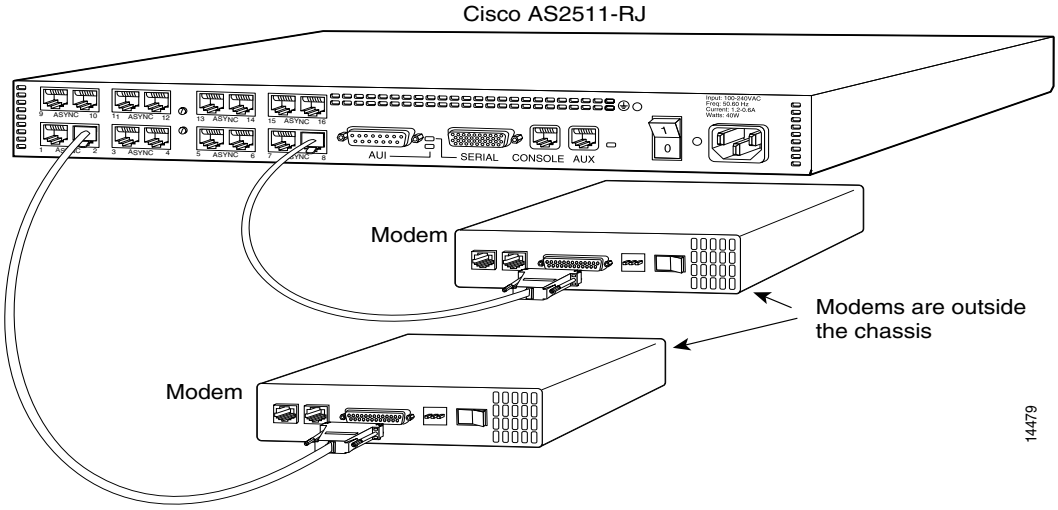
To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

For a complete description of the modem support commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

External Modems on Low-End Access Servers

Some of the Cisco lower-end access servers, such as the Cisco AS2511-RJ shown in Figure 23, have cable connections to external modems. The asynchronous interfaces and lines are inside the access server.

Figure 23 Cisco AS2511-RJ Access Server



When you configure modems to function with your access server, you must provide initialization strings and other settings on the modem to tell it how to function with the access server.

This section assumes that you have already physically attached the modem to the access server. If not, refer to the user guide or installation and configuration guide for your access server for information about attaching modems.

Automatically Configuring an External Modem

The Cisco IOS software can issue initialization strings automatically, in a file called a modemcap, for most types of modems externally attached to the access server. A modemcap is a series of parameter settings that are sent to your modem to configure it to interact with the Cisco device in a specified way. The Cisco IOS software defines modemcaps that have been found to properly initialize most modems so that they function properly with Cisco routers and access servers. For Cisco IOS Release 12.2, these modemcaps have the following names:

- default—Generic Hayes interface external modem
- codex_3260—Motorola Codex 3260 external
- usr_courier—U.S. Robotics Courier external
- usr_sportster—U.S. Robotics Sportster external
- hayes_optima—Hayes Optima external[1]
- global_village—Global Village Teleport external
- viva—Viva (Rockwell ACF with MNP) external
- telebit_t3000—Telebit T3000 external
- nec_v34—NEC V.34 external
- nec_v110—NEC V.110 TA external
- nec_piafs—NEC PIAFS TA external

[1]The hayes_optima modemcap is not recommended for use; instead, use the default modemcap.

Enter these modemcap names with the **modemcap entry** command.

If your modem is not on this list and if you know what modem initialization string you need to use with it, you can create your own modemcap; see the following procedure "Using the Modem Autoconfigure Type Modemcap Feature." To have the Cisco IOS software determine what type of modem you have, use the **modem autoconfigure discovery** command to configure it, as described in the procedure "Using the Modem Autoconfigure Discovery Feature."

Using the Modem Autoconfigure Type Modemcap Feature

Step 1 Use the **modemcap edit** command to define your own modemcap entry.

The following example defines modemcap MODEMCAPNAME:

```
Router(config)# modemcap edit MODEMCAPNAME miscellaneous &FS0=1&D3
```

Step 2 Apply the modemcap to the modem lines as shown in the following example:

```
Router# terminal monitor
Router# debug confmodem
Modem Configuration Database debugging is on
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# line 33 34
Router(config-line)# modem autoconfigure type MODEMCAPNAME
Router(config-line)#
Jan 16 18:12:59.643: TTY34: detection speed (115200) response ---OK---
Jan 16 18:12:59.643: TTY34: Modem command:  --AT&FS0=1&D3--
Jan 16 18:12:59.659: TTY33: detection speed (115200) response ---OK---
Jan 16 18:12:59.659: TTY33: Modem command:  --AT&FS0=1&D3--
Jan 16 18:13:00.227: TTY34: Modem configuration succeeded
Jan 16 18:13:00.227: TTY34: Detected modem speed 115200
Jan 16 18:13:00.227: TTY34: Done with modem configuration
Jan 16 18:13:00.259: TTY33: Modem configuration succeeded
Jan 16 18:13:00.259: TTY33: Detected modem speed 115200
Jan 16 18:13:00.259: TTY33: Done with modem configuration
```

Using the Modem Autoconfigure Discovery Feature

If you prefer the modem software to use its autoconfigure mechanism to configure the modem, use the **modem autoconfigure discovery** command.

The following example shows how to configure modem autoconfigure discovery mode:

```
Router# terminal monitor
Router# debug confmodem
Modem Configuration Database debugging is on
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# line 33 34
Router(config-line)# modem autoconfigure discovery
Jan 16 18:16:17.724: TTY33: detection speed (115200) response ---OK---
Jan 16 18:16:17.724: TTY33: Modem type is default
Jan 16 18:16:17.724: TTY33: Modem command:  --AT&F&C1&D2S0=1H0--
Jan 16 18:16:17.728: TTY34: detection speed (115200) response ---OK---
Jan 16 18:16:17.728: TTY34: Modem type is default
Jan 16 18:16:17.728: TTY34: Modem command:  --AT&F&C1&D2S0=1H0--
Jan 16 18:16:18.324: TTY33: Modem configuration succeeded
```

```
Jan 16 18:16:18.324: TTY33: Detected modem speed 115200
Jan 16 18:16:18.324: TTY33: Done with modem configuration
Jan 16 18:16:18.324: TTY34: Modem configuration succeeded
Jan 16 18:16:18.324: TTY34: Detected modem speed 115200
Jan 16 18:16:18.324: TTY34: Done with modem configuration
```

Manually Configuring an External Modem

If you cannot configure your modem automatically, you must configure it manually. This section describes how to determine and issue the correct initialization string for your modem and how to configure your modem with it.

Modem command sets vary widely. Although most modems use the Hayes command set (prefixing commands with **at**), Hayes-compatible modems do not use identical **at** command sets.

Refer to the documentation that came with your modem to learn how to examine the current and stored configuration of the modem that you are using. Generally, you enter **at** commands such as **&v**, **i4**, or ***o** to view, inspect, or observe the settings.

Timesaver You must first create a direct Telnet or connection session to the modem before you can send an initialization string. You can use **AT&F** as a basic modem initialization string in most cases. To establish a direct Telnet session to an external modem, determine the IP address of your LAN (Ethernet) interface, and then enter a Telnet command to port 2000 + *n* on the access server, where *n* is the line number to which the modem is connected. See the sections "Testing the Modem Connection" and "Managing Telnet Sessions" for more information about making Telnet connections.

A sample modem initialization string for a US Robotics Courier modem is as follows:

```
&b1&h1&r2&c1&d3&m4&k1s0=1
```

Modem initialization strings enable the following functions:

- Locks the speed of the modem to the speed of the serial port on the access server
- Sets hardware flow control (RTS/CTS or request to send/clear to send)
- Ensures correct data carrier detect (DCD) operation
- Ensures proper data terminal ready (DTR) interpretation
- Answers calls on the first ring

Note Make sure to turn off automatic baud rate detection because the modem speeds must be set to a fixed value.

The port speed must not change when a session is negotiated with a remote modem. If the speed of the port on the access server is changed, you must establish a direct Telnet session to the modem and send an **at** command so that the modem can learn the new speed.

Modems differ in the method that they use to lock the EIA/TIA-232 (serial) port speed. In the modem documentation, vendors use terms such as port-rate adjust, speed conversion, or buffered mode. Enabling error correction often puts the modem in the buffered mode. Refer to your modem documentation to learn how your modem locks speed (check the settings **&b**, **\j**, **&q**, **\n**, or s-register settings).

RTS and CTS signals must be used between the modem and the access server to control the flow of data. Incorrectly configuring flow control for software or setting no flow control can result in hung sessions and loss of data. Modems differ in the method that they use to enable hardware flow control. Refer to your modem documentation to learn how to enable hardware flow control (check the settings **&e**, **&k**, **&h**, **&r**, or s-register).

The modem must use the DCD wire to indicate to the access server when a session has been negotiated and is established with a remote modem. Most modems use the setting **&c1**. Refer to your modem documentation for the DCD settings used with your modem.

The modem must interpret a toggle of the DTR signal as a command to drop any active call and return to the stored settings. Most modems use the settings **&d2** or **&d3**. Refer to your modem documentation for the DTR settings used with your modem.

If a modem is used to service incoming calls, it must be configured to answer a call after a specific number of rings. Most modems use the setting **s0=1** to answer the call after one ring. Refer to your modem documentation for the settings used with your modem.

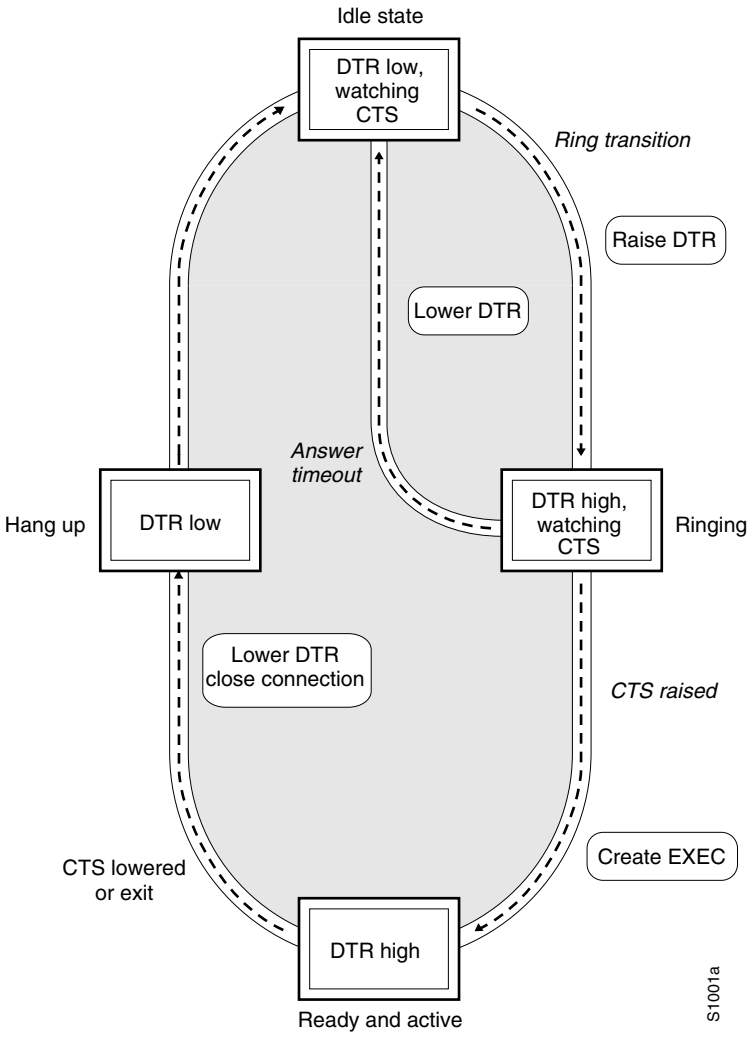
Supporting Dial-In Modems

The Cisco IOS software supports dial-in modems that use DTR to control the off-hook status of the telephone line. This feature is supported primarily on old-style modems, especially those in Europe. To configure the line to support this feature, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem callin`	Configures a line for a dial-in modem.

Figure 24 illustrates the **modem callin** command. When a modem dialing line is idle, it has its DTR signal at a low state and waits for a transition to occur on the data set ready (DSR) input. This transition causes the line to raise the DTR signal and start watching the CTS signal from the modem. After the modem raises CTS, the Cisco IOS software creates an EXEC session on the line. If the timeout interval (set with the **modem answer-timeout** command) passes before the modem raises the CTS signal, the line lowers the DTR signal and returns to the idle state.

Figure 24 EXEC Creation on a Line Configured for Modem Dial-In



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

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

Testing the Modem Connection

To test the connection, send the modem the AT command to request its attention. The modem should respond with "OK." For example:

```
at
OK
```

If the modem does not reply to the **at** command, perform the following steps:

Step 1 Enter the **show users** EXEC command and scan the display output. The output should not indicate that the line is in use. Also verify that the line is configured for **modem inout**.

Step 2 Enter the **show line** EXEC command. The output should contain the following two lines:

```
Modem state: Idle
Modem hardware state: CTS noDSR  DTR RTS
```

If the output displays "no CTS" for the modem hardware state, the modem is not connected, is not powered up, is waiting for data, or might not be configured for hardware flow control.

Step 3 Verify the line speed and modem transmission rate. Make sure that the line speed on the access server matches the transmission rate, as shown in Table 13.

Table 13 Matching Line Speed with Transmission Rate

Modem Transmission Rate (in bits per second)	Line Speed on the Access Server (in bits per second)
9600	38400
14400	57600
28800	115200

To verify the line speed, use the **show run** EXEC command. The line configuration fragment appears at the tail end of the output.

The following example shows that lines 7 through 9 are transmitting at 115200 bits per second (bps). Sixteen 28800-kbps modems are connected to a Cisco AS2511-RJ access server via a modem cable.

```
Router# show run

Building configuration...

Current configuration:
. . .
!
 line 1 16
  login local
  modem InOut
  speed 115200
  transport input all
  flowcontrol hardware
  script callback callback
  autoselect ppp
  autoselect during-login
```

Step 4 The speeds of the modem and the access server are likely to be different. If so, switch off the modem, and then switch it back on. This action should change the speed of the modem to match the speed of the access server.

Step 5 Check your cabling and the modem configuration (echo or result codes might be off). Enter the appropriate **at** modem command to view the modem configuration, or use the **at&f** command to return to factory defaults. Refer to your modem documentation to learn the appropriate **at** command to view your modem configuration.

Note See the section "Configuring Cisco Integrated Modems Using Modem Attention Commands" in the "Configuring and Managing Integrated Modems" chapter for information about modem attention commands for the Cisco internal modems.

Managing Telnet Sessions

You communicate with an external modem by establishing a direct Telnet session from the asynchronous line on the access server, which is connected to the modem. This process is also referred to as *reverse Telnet*. Performing a reverse Telnet means that you are initiating a Telnet session out the asynchronous line, instead of accepting a connection into the line (called a *forward* connection).

Note Before attempting to allow inbound connections, make sure that you close all open connections to the modems attached to the access server. If you have a modem port in use, the modem will not accept a call properly.

To establish a direct Telnet session to an external modem, determine the IP address of your LAN (Ethernet) interface, and then enter a Telnet command to port 2000 + *n* on the access server, where *n* is the line number to which the modem is connected. For example, to connect to the modem attached to line 1, enter the following command from an EXEC session on the access server:

```
Router# telnet 172.16.1.10 2001
Trying 172.16.1.10, 2001 ... Open
```

This example enables you to communicate with the modem on line 1 using the AT (attention) command set defined by the modem vendor.

Timesaver Use the **ip host** configuration command to simplify direct Telnet sessions with modems. The **ip host** command maps an IP address of a port to a device name. For example, the **modem1 2001 172.16.1.10** command enables you to enter **modem1** to initiate a connection with the modem, instead of repeatedly entering **telnet 172.16.1.10 2001** each time you want to communicate with the modem.

You can also configure asynchronous rotary line queueing, which places Telnet login requests in a queue when lines are busy. See the section "Configuring Asynchronous Rotary Line Queueing" in the "Configuring Asynchronous Lines and Interfaces" chapter for more information.

Suspending Telnet Sessions:

When you are connected to an external modem, the direct Telnet session must be terminated before the line can accept incoming calls. If you do not terminate the session, it will be indicated in the output of the **show users** command and will return a modem state of ready if the line is still in use. If the line is no longer in use, the output of the **show line** *value* command will return a state of idle. Terminating the Telnet session requires first suspending it, then disconnecting it.

To suspend a Telnet session, perform the following steps:

Step 1 Enter Ctrl-Shift-6 x to suspend the Telnet session:

```
- suspend keystroke -
Router#
```

Note Ensure that you can reliably issue the escape sequence to suspend a Telnet session. Some terminal emulation packages have difficulty sending the Ctrl-Shift-6 x sequence. Refer to your terminal emulation documentation for more information about escape sequences.

Step 2 Enter the **where** EXEC command to check the connection numbers of open sessions:

```
Router# where
Conn Host                Address             Byte  Idle Conn Name
*  1 172.16.1.10         172.16.1.10            0     0  172.16.1.10
   2 172.16.1.11         172.16.1.11            0    12  modem2
```

Step 3 When you have suspended a session with one modem, you can connect to another modem and suspend it:

```
Router# telnet modem2
Trying modem2 (172.16.1.11, 2002) ... Open

- suspend keystroke -
Router#
```

Step 4 To disconnect (completely close) a Telnet session, enter the **disconnect** EXEC command:

```
Router# disconnect line 1
Closing connection to 172.16.1.10 [confirm] y
Router# disconnect line 2
Closing connection to 172.16.1.11 [confirm] y
Router#
```

Modem Troubleshooting Tips

Table 14 contains troubleshooting tips on modem access and control.

Table 14 Modem Troubleshooting Tips

Problem	Likely Cause
Connection refused.	Someone already has a connection to that port. or an EXEC is running on that port. or The modem failed to lower the carrier detect (CD) signal after a call disconnected, resulting in an EXEC that remained active after disconnect. To force the line back into an idle state, clear the line from the console and try again. If it still fails, ensure that you have set **modem inout** command for that line. If you don't have modem control, either turn off EXEC on the line (by using the **exec-timeout** line configuration command) before making a reverse connection or configure the modem using an external terminal. As a last resort, disconnect the modem, clear the line, make the Telnet connection, and then attach the modem. The prevents a misconfigured modem from denying you line access.
Connection appears to hang.	Try entering "^U" (clear line), "^Q" (XON), and press Return a few times to try to establish terminal control.
EXEC does not come up; autoselect is on.	Press Return to enter EXEC.
Modem does not hang up after entering **quit**.	The modem is not receiving DTR information, or you have not set up modem control on the router.
Interrupts another user session when you dial in.	The modem is not dropping CD on disconnect, or you have not set up modem control on the router.
Connection hangs after entering "+++" on the dialing modem, followed by an ATO.	The answering modem saw and interpreted the "+++" when it was echoed to you. This is a bug in the answering modem, common to many modems. There may be a switch to work around this problem; check the modem's documentation.
Losing data.	You may have Hardware Flow Control only on for either the router's line (DTE) or the modem (DCE). Hardware Flow Control should be on for both or off for both, but not for only one.
Using MDCE.	Turn MDCE into an MMOD by moving pin 6 to pin 8 because most modems use CD and not DSR to indicate the presence of carrier. You can also program some modems to provide carrier info via DSR.

Checking Other Modem Settings

This section defines other settings that might be needed or desirable, depending on your modem.

Error correction can be negotiated between two modems to ensure a reliable data link. Error correction standards include Link Access Procedure for Modems (LAPM) and MNP4. V.42 error correction allows either LAPM or MNP4 error correction to be negotiated. Modems differ in the way they enable error correction. Refer to your modem documentation for the error correction methods used with your modem.

Data compression can be negotiated between two modems to allow for greater data throughput. Data compression standards include V.42*bis* and MNP5. Modems differ in the way they enable data compression. Refer to your modem documentation for the data compression settings used with your modem.

Modem Signal and Line States

This chapter describes modem states in the following section:

- Signal and Line State Diagrams

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

For a complete description of the modem support commands in this chapter, refer to the *Cisco IOS Modem Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Signal and Line State Diagrams

The following signal and line state diagrams accompany some of the tasks in the following sections to illustrate how the modem control works:

- Configuring Automatic Dialing
- Automatically Answering a Modem
- Supporting Dial-In and Dial-Out Connections
- Configuring a Line Timeout Interval
- Closing Modem Connections
- Configuring a Line to Disconnect Automatically
- Supporting Reverse Modem Connections and Preventing Incoming Calls

The diagrams show two processes:

- The “create daemon” process creates a tty daemon that handles the incoming network connection.
- The “create EXEC” process creates the process that interprets user commands. (See Figure 25 through Figure 29.)

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

Figure 25 illustrates line states when no modem control is set. The DTR output is always high, and CTS and RING are completely ignored. The Cisco IOS software starts an EXEC session when the user types the activation character. Incoming TCP connections occur instantly if the line is not in use and can be closed only by the remote host.

Figure 25 EXEC and Daemon Creation on a Line with No Modem Control

Configuring Automatic Dialing

With the dialup capability, you can set a modem to dial the phone number of a remote router automatically. This feature offers cost savings because phone line connections are made only when they are needed—you pay for using the phone line only when there is data to be received or sent.

To configure a line for automatic dialing, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem dtr-active`	Configures a line to initiate automatic dialing.

Using the **modem dtr-active** command causes a line to raise DTR signal only when there is an outgoing connection (such as reverse Telnet, NetWare Asynchronous Support Interface (NASI), or DDR), rather than leave DTR raised all the time. When raised, DTR potentially tells the modem that the router is ready to accept a call.

Automatically Answering a Modem

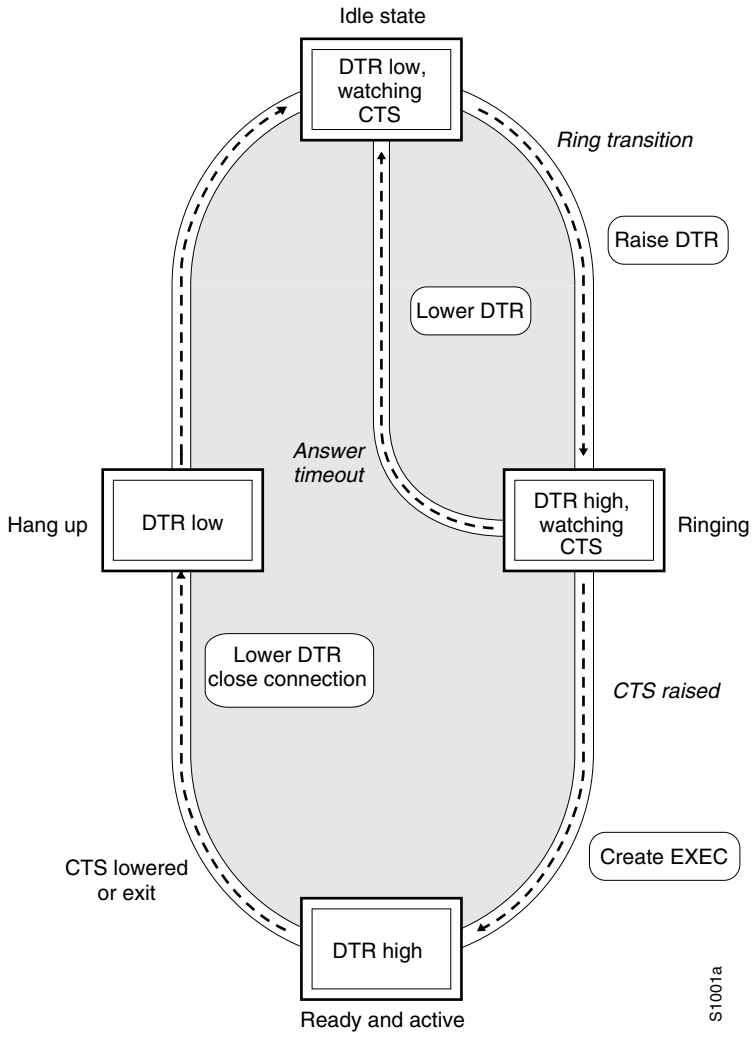
You can configure a line to answer a modem automatically. You also can configure the modem to answer the telephone on its own (as long as DTR is high), drop connections when DTR is low, and use its Carrier Detect (CD) signal to accurately reflect the presence of carrier. (Configuring the modem is a modem-dependent process.) First, wire the modem CD signal (generally pin-8) to the router RING input (pin-22), then use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem dialin`	Configures a line to automatically answer a modem.

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

Figure 26 illustrates the **modem dialin** process with a high-speed dialup modem. When the Cisco IOS software detects a signal on the RING input of an idle line, it starts an EXEC or autobaud process on that line. If the RING signal disappears on an active line, the Cisco IOS software closes any open network connections and terminates the EXEC facility. If the user exits the EXEC or the software terminates because of no user input, the line makes the modem hang up by lowering the DTR signal for 5 seconds. After 5 seconds, the modem is ready to accept another call.

Figure 26 EXEC Creation on a Line Configured for a High-Speed Modem



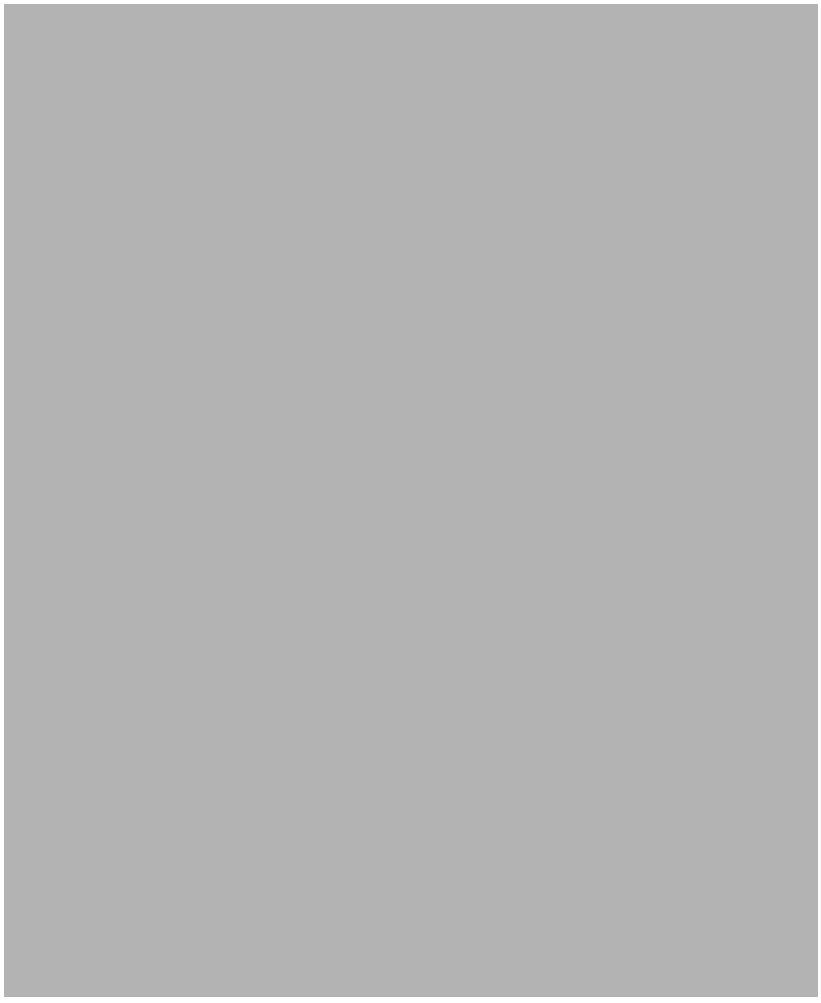
Supporting Dial-In and Dial-Out Connections

To configure a line for both incoming and outgoing calls, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem inout`	Configures a line for both incoming and outgoing calls.

Figure 27 illustrates the **modem inout** command. If the line is activated by raising the data set ready (DSR) signal, it functions exactly as a line configured with the **modem dialin** line configuration command described in the section "Automatically Answering a Modem" earlier in this chapter. If the line is activated by an incoming TCP connection, the line functions similarly to lines not used with modems.

Figure 27 EXEC and Daemon Creation for Incoming and Outgoing Calls



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

Configuring a Line Timeout Interval

To change the interval that the Cisco IOS software waits for the CTS signal after raising the DTR signal in response to the DSR (the default is 15 seconds), use the following command in line configuration mode. The timeout applies to the **modem callin** command only.

Command	Purpose
`Router(config-line)# modem answer-timeout seconds`	Configures modem line timing.

Note The DSR signal is called RING on older ASM-style chassis.

Closing Modem Connections

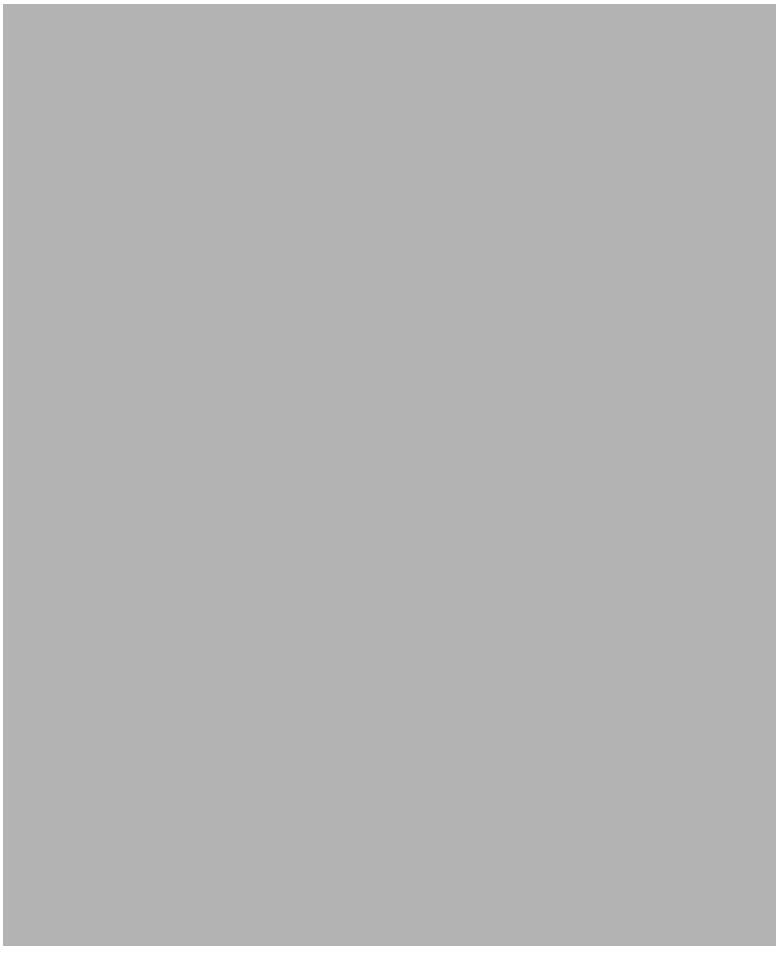
Note The **modem cts-required** command was replaced by the **modem printer** command in Cisco IOS Release 12.2.

To configure a line to close connections from a user's terminal when the terminal is turned off and to prevent inbound connections to devices that are out of service, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem cts-required`	Configures a line to close connections.

Figure 28 illustrates the **modem cts-required** command operating in the context of a continuous CTS signal. This form of modem control requires that the CTS signal be high for the entire session. If CTS is not high, the user input is ignored and incoming connections are refused (or sent to the next line in a rotary group).

Figure 28 EXEC and Daemon Creation on a Line Configured for Continuous CTS



Configuring a Line to Disconnect Automatically

To configure automatic line disconnect, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# autohangup`	Configures automatic line disconnect.

The **autohangup** command causes the EXEC facility to issue the **exit** command when the last connection closes. This feature is useful for UNIX-to-UNIX copy program (UUCP) applications because UUCP scripts cannot issue a command to hang up the telephone. This feature is not used often.

Supporting Reverse Modem Connections and Preventing Incoming Calls

In addition to initiating connections, the Cisco IOS software can receive incoming connections. This capability allows you to attach serial and parallel printers, modems, and other shared peripherals to the router or access server and drive them remotely from other modem-connected systems. The Cisco IOS software supports reverse TCP, XRemote, and local-area transport (LAT) connections.

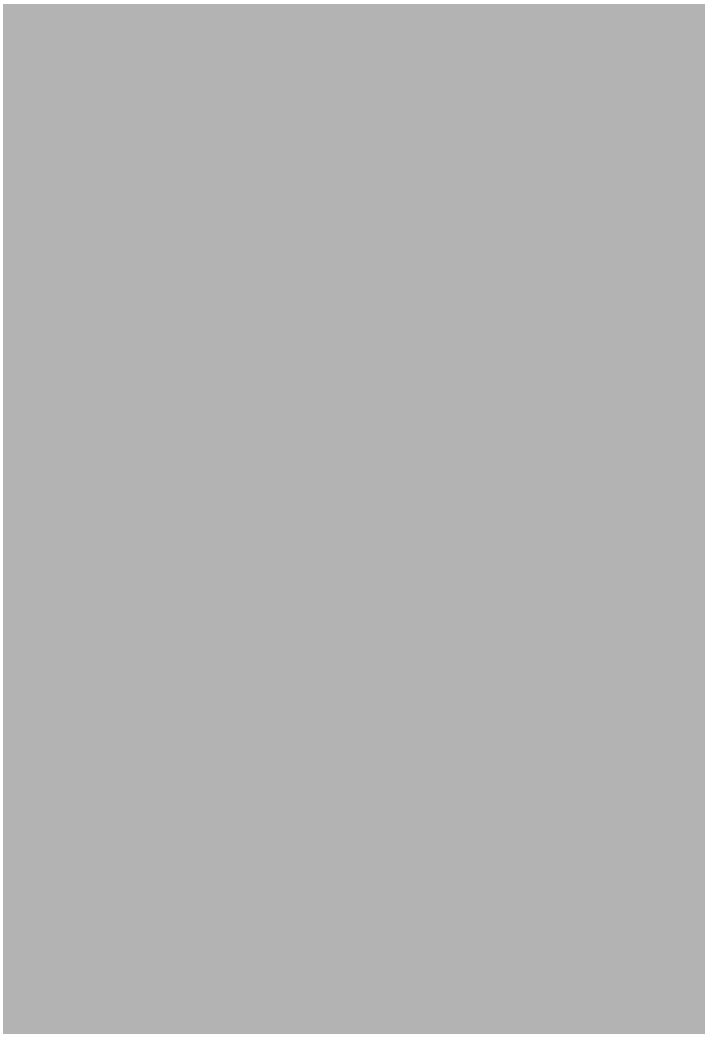
The specific TCP port or socket to which you attach the device determines the type of service that the Cisco IOS software provides on a line. When you attach the serial lines of a computer system or a data terminal switch to the serial lines of the access server, the access server can act as a network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

The Cisco IOS software supports ports connected to computers that are connected to modems. To configure the Cisco IOS software to function somewhat like a modem, use the following command in line configuration mode. This command also prevents incoming calls.

Command	Purpose
`Router(config-line)# modem callout`	Configures a line for reverse connections and prevents incoming calls.

Figure 29 illustrates the **modem callout** process. When the Cisco IOS software receives an incoming connection, it raises the DTR signal and waits to see if the CTS signal is raised to indicate that the host has noticed the router DTR signal. If the host does not respond within the interval set by the **modem answer-timeout** line configuration command, the software lowers the DTR signal and drops the connection.

Figure 29 Daemon Creation on a Line Configured for Modem Dial-Out



Creating and Using Modem Chat Scripts

This chapter describes how to create and use modem chat scripts. These tasks are presented in the following main sections:

- Chat Script Overview
- How To Configure Chat Scripts
- Using Chat Scripts

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

For a complete description of the modem support commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference* publication. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Chat Script Overview

Chat scripts are strings of text used to send commands for modem dialing, logging in to remote systems, and initializing asynchronous devices connected to an asynchronous line.

Note On a router, chat scripts can be configured only on the auxiliary port.

A chat script must be configured to dial out on asynchronous lines. You also can configure chat scripts so that they can be executed automatically for other specific events on a line, or so that they are executed manually.

Each chat script is defined for a different event. These events can include the following:

- Line activation
- Incoming connection initiation
- Asynchronous dial-on-demand routing (DDR)
- Line resets
- Startup

Note Outbound chat scripts are not supported on lines where modem control is set for inbound activity only using the **modem dialin** command.

How To Configure Chat Scripts

The following tasks must be performed before a chat script can be used:

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

To configure a chat script, perform the tasks in the following sections:

- Understanding Chat Script Naming Conventions (Required)
- Creating a Chat Script (Required)
- Configuring the Line to Activate Chat Scripts (Required)
- Manually Testing a Chat Script on an Asynchronous Line (Optional)

See the section "Using Chat Scripts" later in this chapter for examples of how to use chat scripts.

Understanding Chat Script Naming Conventions

When you create a script name, include the modem vendor, type, and modulation, separated by hyphens, as follows:

vendor-type-modulation

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

```
telebit-t3000-v32bis
```

Note Adhering to the recommended naming convention allows you to specify a range of chat scripts by using partial names in UNIX-style regular expressions. The regular expressions are used to match patterns and select chat scripts to use. This method is particularly useful for dialer rotary groups on an interface that dials multiple destinations. Regular expressions are described in the "Regular Expressions" appendix in the *Cisco IOS Terminal Services Configuration Guide*.

Creating a Chat Script

We recommend that one chat script (a "modem" chat script) be written for placing a call and that another chat script (a "system" or "login" chat script) be written to log in to remote systems, where required.

To define a chat script, use the following command in global configuration mode:

Command	Purpose
`Router(config)# chat-script` *script-name expect send...*	Creates a script that will place a call on a modem, log in to a remote system, or initialize an asynchronous device on a line.

The Cisco IOS software waits for the string from the modem (defined by the *expect* portion of the script) and uses it to determine what to send back to the modem (defined by the *send* portion of the script).

Chat String Escape Key Sequences

Chat script send strings can include the special escape sequences listed in Table 15.

Table 15 Chat Script Send String Escape Sequences

Escape Sequence	Description
\	Sends the ASCII character with its octal value.
\\	Sends a backslash (\) character.
\”	Sends a double-quote (“) character (does not work *within* double quotes).
\c	Suppresses a new line at the end of the send string.
\d	Delays for 2 seconds.
\K	Inserts a BREAK.
\n	Sends a newline or linefeed character.
\N	Sends a null character.
\p	Pauses for 0.25 second.
\q	Reserved, not yet used.
\r	Sends a return.
\s	Sends a space character.
\t	Sends a tab character.
\T	Replaced by phone number.
“ ”	Expects a null string.
BREAK	Causes a BREAK. This sequence is sometimes simulated with line speed changes and null characters. May not work on all systems.
EOT	Sends an end-of-transmission character.

Adding a Return Key Sequence

After the connection is established and you press the Return key, you must often press Return a second time before the prompt appears. To create a chat script that enters this additional Return key for you, include the following string with the Return key escape sequence (see Table 15) as part of your chat script:

```
ssword:-/r-ssword
```

This part of the script specifies that, after the connection is established, you want **ssword** to be displayed. If it is not displayed, you must press Return again after the timeout passes. (For more information about expressing characters in chat scripts, see the "Regular Expressions" appendix in the *Cisco IOS Terminal Services Configuration Guide*.)

Chat String Special-Case Script Modifiers

Special-case script modifiers are also supported; refer to Table 16 for examples.

Table 16 Special-Case Script Modifiers

Special Case	Function
ABORT *string*	Designates a string whose presence in the input indicates that the chat script has failed. (You can have as many active abort entries as you like.)
TIMEOUT *time*	Sets the time to wait for input, in seconds. The default is 5 seconds, and a timeout of 60 seconds is recommended for V.90 modems.

For example, if a modem reports BUSY when the number dialed is busy, you can indicate that you want the attempt stopped at this point by including ABORT BUSY in your chat script.

Note If you use the *expect-send* pair ABORT SINK instead of ABORT ERROR, the system terminates abnormally when it encounters SINK instead of ERROR.

Configuring the Line to Activate Chat Scripts

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

To define a chat script to start automatically when a specific event occurs, use one of the following commands in line configuration mode:

Command	Purpose
`Router(config-line)# script activation` *regexp*[1]	Starts a chat script on a line when the line is activated (every time a command EXEC is started on the line).
`Router(config-line)# script connection` *regexp*	Starts a chat script on a line when a network connection is made to the line.
`Router(config-line)# script dialer` *regexp*	Specifies a modem script for DDR on a line.
`Router(config-line)# script reset` *regexp*[2]	Starts a chat script on a line whenever the line is reset.
`Router(config-line)# script startup` *regexp*[2]	Starts a chat script on a line whenever the system is started up.

1. The *regexp* argument is a regular expression that is matched to a script name that has already been defined using the **chat-script** command.
2. Do not use the **script reset** or **script startup** commands to configure a modem; instead use the **modem autoconfigure** command.

Note Outbound chat scripts are not supported on lines where modem control is set for inbound activity only (using the **modem dialin** command).

Manually Testing a Chat Script on an Asynchronous Line

To test a chat script on any line that is currently not active, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# debug chat line number`	Starts detailed debugging on the specified line.
Step 2	`Router# start-chat regexp [line-number [dialer-string]]`	Starts a chat script on any asynchronous line.

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

Using Chat Scripts

The following sections provide examples of how to use chat scripts:

- Generic Chat Script Example
- Traffic-Handling Chat Script Example
- Modem-Specific Chat Script Examples
- Dialer Mapping Example
- System Login Scripts and Modem Script Examples

Generic Chat Script Example

The following example chat script includes a pair of empty quotation marks (“ ”), which means “expect anything,” and \r, which means “send a return”:

```
" " \r "name:" "myname" "ord":" "mypassword" ">" "slip default"
```

Traffic-Handling Chat Script Example

The following example shows a configuration in which, when there is traffic, a random line will be used. The dialer code will try to find a script that matches either the modem script *.*-v32* or the system script cisco. If there is no match for either the modem script or the system script, you will see a “no matching chat script found” message.

```
interface dialer 1
! v.32 rotaries are in rotary 1.
dialer rotary-group 1
! Use v.32 generic script.
dialer map ip 10.0.0.1 modem-script .*-v32 system-script cisco 1234
```

Modem-Specific Chat Script Examples

The following example shows line chat scripts being specified for lines connected to Telebit and US Robotics modems:

```
! Some lines have Telebit modems.
line 1 6
 script dialer telebit.*
! Some lines have US Robotics modems.
line 7 12
 script dialer usr.*
```

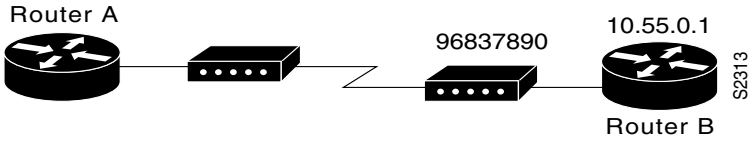
Dialer Mapping Example

The following example shows a modem chat script called dial and a system login chat script called login:

```
chat-script dial ABORT ERROR "" "AT Z" OK "ATDT \T" TIMEOUT 60 CONNECT \c
chat-script login ABORT invalid TIMEOUT 60 name: myname word: mypassword ">" "slip
default"
interface async 10
 dialer in-band
 dialer map ip 10.55.0.1 modem-script dial system-script login 96837890
```

Figure 30 illustrates the configuration.

Figure 30 Chat Script Configuration and Function



- The configuration is on Router A.
- The modem chat script dial is used to dial out to the modem at Router B.
- The system login chat script login is used to log in to Router B.
- The phone number is the number of the modem attached to Router B.
- The IP address in the **dialer map** command is the address of Router B.

In the sample script shown, the **dialer in-band** command enables DDR on asynchronous interface 10, and the **dialer map** command dials 96837890 after finding the specified dialing and the system login scripts. When a packet is received for 10.55.0.1, the first thing to happen is that the modem script is implemented. Table 17 lists the functions that are implemented with each expect-send pair in the modem script called dial.

Table 17 Example Modem Script Execution

Expect and Send Pair	Implementation
ABORT ERROR	Ends the script execution if the text “ERROR” is found. (You can have as many active abort entries as you like.)
“ ” “AT Z”	Without expecting anything, sends an “AT Z” command to the modem. (Note the use of quotation marks to allow a space in the send string.)
OK “ATDT \T	Waits to see “OK.” Sends “ATDT 96837890.”
TIMEOUT 60	Waits up to 60 seconds for next expect string.
CONNECT \c	Expects “connect,” but does not send anything. (Note that *\c* is effectively nothing; “ ” would have indicated nothing followed by a carriage return.)

After the modem script is successfully executed, the system login script is executed. Table 18 lists the functions that are executed with each expect-send pair in the system script called login.

Table 18 Example System Script Execution

Expect and Send Pair	Implementation
ABORT invalid	Ends the script execution if the message “invalid username or password” is displayed.
TIMEOUT 60	Waits up to 60 seconds.
name: *username*	Waits for “name:” and sends username. (Using just “name:” will help avoid any capitalization issues.)
word: *password*	Waits for “word:” and sends the password.
“>” “slip default”	Waits for the > prompt and places the line into Serial Line Internet Protocol (SLIP) mode with its default address.

System Login Scripts and Modem Script Examples

The following example shows the use of chat scripts implemented with the **system-script** and **modem-script** options of the **dialer map** command.

If there is traffic for IP address 10.2.3.4, the router will dial the 91800 number using the usrobotics-v32 script, matching the regular expression in the modem chat script. Then the router will run the unix-slip chat script as the system script to log in.

If there is traffic for 10.3.2.1, the router will dial 8899 using usrobotics-v32, matching both the modem script and modem chat script regular expressions. The router will then log in using the cisco-compressed script.

```
! Script for dialing a usr v.32 modem:
chat-script usrobotics-v32 ABORT ERROR "" "AT Z" OK "ATDT \T" TIMEOUT 60 CONNECT \c
!
! Script for logging into a UNIX system and starting up SLIP:
chat-script unix-slip ABORT invalid TIMEOUT 60 name: billw word: wewpass ">" "slip
default"
!
```

```
! Script for logging into a Cisco access server and starting up TCP header compression:
chat-script cisco-compressed...
!
line 15
 script dialer usrobotics-*
!
interface async 15
 dialer map ip 10.2.3.4 system-script *-v32 system-script cisco-compressed 91800
 dialer map ip 10.3.2.1 modem-script *-v32 modem-script cisco-compressed 91800
```

Configuring Dial Backup with Dialer Profiles

This chapter describes how to configure dialer interfaces, which can be configured as the logical intermediary between one or more physical interfaces and another physical interface that is to function as backup. It includes the following main sections:

- Dial Backup with Dialer Profiles Overview
- How to Configure Dial Backup with Dialer Profiles
- Configuration Example of Dialer Profile for ISDN BRI Backing Up Two Leased Lines

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

For a complete description of the dial backup commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Dial Backup with Dialer Profiles Overview

A backup interface is an interface that stays idle until certain circumstances occur; then it is activated. Dialer interfaces can be configured to use a specific dialing pool; in turn, physical interfaces can be configured to belong to the same dialing pool.

See the section "Configuration Example of Dialer Profile for ISDN BRI Backing Up Two Leased Lines" at the end of this chapter for a comprehensive example of a dial backup interface using dialer profiles. In the example, one BRI functions as backup to two serial lines and can make calls to two different destinations.

How to Configure Dial Backup with Dialer Profiles

To configure a dialer interface and a specific physical interface to function as backup to other physical interfaces, perform the tasks in the following sections:

- Configuring a Dialer Interface (Required)
- Configuring a Physical Interface to Function As Backup (Required)
- Configuring Interfaces to Use a Backup Interface (Required)

Configuring a Dialer Interface

To configure the dialer interface that will be used as an intermediary between a physical interface that will function as backup interface and the interfaces that will use the backup, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface dialer number`	Creates a dialer interface and begins interface configuration mode.
Step 2	`Router(config-if)# ip unnumbered loopback0`	Specifies IP unnumbered loopback.
Step 3	`Router(config-if)# encapsulation ppp`	Specifies PPP encapsulation.
Step 4	`Router(config-if)# dialer remote-name username`	Specifies the Challenge Handshake Authentication Protocol (CHAP) authentication name of the remote router.
Step 5	`Router(config-if)# dialer string dial-string`	Specifies the remote destination to call.
Step 6	`Router(config-if)# dialer pool number`	Specifies the dialing pool to use for calls to this destination.
Step 7	`Router(config-if)# dialer-group group-number`	Assigns the dialer interface to a dialer group.

Configuring a Physical Interface to Function As Backup

To configure the physical interface that is to function as backup, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface and begins interface configuration mode.
Step 2	`Router(config-if)# encapsulation ppp`	Specifies PPP encapsulation.
Step 3	`Router(config-if)# dialer pool-member number`	Makes the interface a member of the dialing pool that the dialer interface will use; make sure the *number* arguments have the same value.
Step 4	`Router(config-if)# ppp authentication chap`	Specifies CHAP authentication.

Configuring Interfaces to Use a Backup Interface

To configure one or more interfaces to use a backup interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface to be backed up and begins interface configuration mode.
Step 2	`Router(config-if)# ip unnumbered loopback0`	Specifies IP unnumbered loopback.

	Command	Purpose
Step 3	`Router(config-if)# backup interface dialer` *number*	Specifies the backup interface and begins interface configuration mode.
Step 4	`Router(config-if)# backup delay` *enable-delay disable-delay*	Specifies delay between the physical interface going down and the backup being enabled, and between the physical interface coming back up and the backup being disabled.

Configuration Example of Dialer Profile for ISDN BRI Backing Up Two Leased Lines

The following example shows the configuration of a site that backs up two leased lines using one BRI. Two dialer interfaces are defined. Each serial (leased line) interface is configured to use one of the dialer interfaces as a backup. Both of the dialer interfaces use dialer pool 1, which has physical interface BRI 0 as a member. Thus, physical interface BRI 0 can back up two different serial interfaces and can make calls to two different sites.

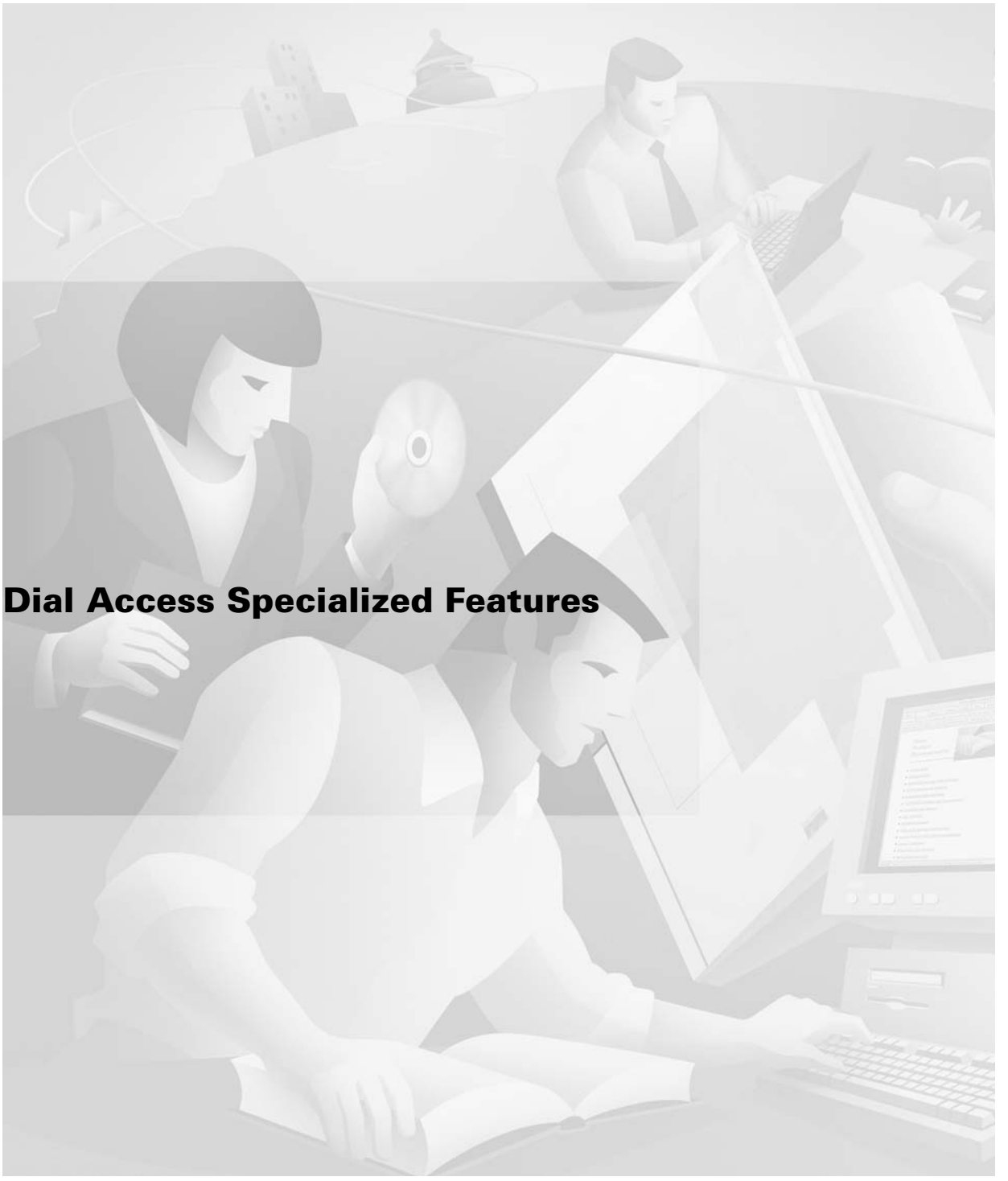
```
interface dialer0
 ip unnumbered loopback0
 encapsulation ppp
 dialer remote-name Remote0
 dialer pool 1
 dialer string 5551212
 dialer-group 1

interface dialer1
 ip unnumbered loopback0
 encapsulation ppp
 dialer remote-name Remote1
 dialer pool 1
 dialer string 5551234
 dialer-group 1

interface bri 0
 encapsulation PPP
 dialer pool-member 1
 ppp authentication chap

interface serial 0
 ip unnumbered loopback0
 backup interface dialer 0
 backup delay 5 10

interface serial 1
 ip unnumbered loopback0
 backup interface dialer1
 backup delay 5 10
```



Dial Access Specialized Features

Configuring per-User Configuration

This chapter describes per-user configuration, a large-scale dial solution. It includes the following main sections:

- Per-User Configuration Overview
- How to Configure a AAA Server for Per-User Configuration
- Monitoring and Debugging Per-User Configuration Settings
- Configuration Examples for Per-User Configuration

This set of features is supported on all platforms that support Multilink PPP (MLP).

A virtual access interface created dynamically for any user dial-in session is deleted when the session ends. The resources used during the session are returned for other dial-in uses.

When a specific user dials in to a router, the use of a per-user configuration from an authentication, authorization, and accounting (AAA) server requires that AAA is configured on the router and that a configuration for that user exists on the AAA server.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

For a complete description of the commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12.2 and the *Cisco IOS Security Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Per-User Configuration Overview

Per-user configuration provides a flexible, scalable, easily maintained solution for customers with a large number of dial-in users. This solution can tie together the following dial-in features:

- Virtual template interfaces, generic interface configuration and router-specific configuration information stored in the form of a virtual template interface that can be applied (*cloned*) to a virtual access interface each time any user dials in. This configuration is described in the chapter "Configuring Virtual Template Interfaces" in this publication.
- AAA per-user security and interface configuration information stored on a separate AAA server and sent by the AAA server to the access server or router in response to authorization requests during the PPP authentication phase. The per-user configuration information can add to or override the generic configuration on a virtual interface.

- Virtual profiles, which can use either or both of the two sources of information listed in the previous bullets for virtual interface configuration. When a user dials in, virtual profiles can apply the generic interface configuration and then apply the per-user configuration to create a unique virtual access interface for that user. This configuration is described in the chapter "Configuring Virtual Profiles" in this publication.

The per-user configuration feature provides these benefits:

- Maintenance ease for service providers with a large number of access servers and a very large number of dial-in users. Service providers need not update all their routers and access servers when user-specific information changes; instead, they can update one AAA server.
- Scalability. By separating generic virtual interface configuration on the router from the configuration for each individual, Internet service providers and other enterprises with large numbers of dial-in users can provide a uniquely configured interface for each individual user. In addition, by separating the generic virtual interface configuration from the physical interfaces on the router, the number and types of physical interfaces on the router or access server are not intrinsic barriers to growth.

General Operational Processes

In general, the per-user configuration process on the Cisco router or network access server proceeds as follows:

1. The user dials in.
2. The authentication and authorization phases occur.
   a. If AAA is configured, the router sends an authorization request to the AAA server.
   b. If the AAA server has information (attribute-value or AV pairs, or other configuration parameters) that defines a configuration for the specific user, the server includes it in the information in the approval response packet.

      Figure 98 illustrates the request and response part of the process that happens when a user dials in, given that AAA is configured and that the AAA server has per-user configuration information for the dial-in user.
   c. The router looks for AV pairs in the AAA approval response.
   d. The router caches the configuration parameters.

Note TACACS servers treat authentication and authorization as two phases; RADIUS servers combine authentication and authorization into a single step. For more detailed information, refer to your server documentation.

Figure 98 Per-User Configuration Authentication and Authorization

3. A virtual access interface is created for this user.
   a. The router finds the virtual template that is set up for virtual profiles, if any, and applies the commands to the virtual access interface.
   b. The router looks for the AV pairs to apply to this virtual access interface to configure it for the dial-in user.
   c. The AV pairs are sent to the Cisco IOS command-line parser, which interprets them as configuration commands and applies them to configure this virtual access interface.

The result of this process is a virtual access interface configured uniquely for the dial-in user.

When the user ends the call, the virtual access interface is deleted and its resources are returned for other dial-in uses.

Note The use of virtual profiles can modify the process that occurs between the user dial-in and the use of AAA configuration information. For more information, see the chapter "Configuring Virtual Profiles" in this publication.

Operational Processes with IP Address Pooling

During IP Control Protocol (IPCP) address negotiation, if an IP pool name is specified for a user, the network access server checks whether the named pool is defined locally. If it is, no special action is required and the pool is consulted for an IP address.

If the required pool is not present (either in the local configuration or as a result of a previous download operation), an authorization call to obtain it is made using the special username:

```
pools-nas-name
```

where *nas-name* is the configured name of the network access server. In response, the AAA server downloads the configuration of the required pool.

This pool username can be changed using Cisco IOS configuration, for example:

```
aaa configuration config-name nas1-pools-definition.cisco.us
```

This command has the effect of changing the username that is used to download the pool definitions from the default name "pools-*nas-name*" to "nas1-pools-definition.cisco.com."

On a TACACS+ server, the entries for an IP address pool and a user of the pool might be as follows:

```
user = nas1-pools {
     service = ppp protocol = ip {
        pool-def#1 = "aaa 10.0.0.1 10.0.0.3"
        pool-def#2 = "bbb 10.1.0.1 10.1.0.10"
        pool-def#3 = "ccc 10.2.0.1 10.2.0.20"
        pool-timeout=60
    }
}

user = georgia {
    login = cleartext lab
    service = ppp protocol = ip {
                  addr-pool=bbb
    }
}
```

On a RADIUS server, the entries for the same IP address pool and user would be as follows:

```
nas1-pools      Password = "cisco" User-Service-Type=Outbound-User
        cisco-avpair = "ip:pool-def#1=aaa 10.0.0.1 10.0.0.3",
        cisco-avpair = "ip:pool-def#2=bbb 10.1.0.1 10.1.0.10",
        cisco-avpair = "ip:pool-def#3=ccc 10.2.0.1 10.2.0.20",
        cisco-avpair = "ip:pool-timeout=60"

georgia Password = "lab"
        User-Service-Type = Framed-User,
        Framed-Protocol = PPP,
                cisco-avpair = "ip:addr-pool=bbb"
```

Note This entry specifies a User-Service-Type of Outbound-User. This attribute is supplied by the network access server to prevent ordinary logins from using the well-known username and password combination of nas1-pools/cisco.

Pools downloaded to a Cisco network access server are not retained in nonvolatile memory and automatically disappear whenever the access server or router restarts. Downloaded pools can also be made to time out automatically by adding a suitable AV pair. For more information, see the section "Supported Attrubutes for AV Pairs" and the pool-timeout attribute in Table 37. Downloaded pools are marked as *dynamic* in the output of the **show ip local pool** command.

Deleting Downloaded Pools

To delete downloaded pools, you can do either of the following:

- Manually delete the definition from the network access server. For example, if "bbb" is the name of a downloaded pool, you can enter the Cisco IOS **no ip local pool bbb** command.

  Deleting a pool definition does not interrupt service for current users. If a pool is deleted and then redefined to include a pool address that is currently allocated, the new pool understands and tracks the address as expected.

- Set an AV pair pool-timeout value; this is a more desirable solution.

  The pool-timeout AV pair starts a timer when the pool is downloaded. Once the timer expires, the pools are deleted. The next reference to the pools again causes an authorization call to be made, and the pool definition is downloaded again. This method allows definitions to be made and changed on the AAA server and propagated to network access servers.

Supported Attributes for AV Pairs

Table 37 provides a partial list of the Cisco-specific supported attributes for AV pairs that can be used for per-user virtual interface configuration. For complete lists of Cisco-specific, vendor-specific, and TACACS+ supported attributes, see the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference*.

Table 37 Partial List of Cisco-Specific Supported AV Pair Attributes

Attribute	Meaning
inacl#	An input access list definition. For IP, standard or extended access list syntax can be used, although you cannot mix them within a single list. For Internet Protocol Exchange (IPX), only extended syntax is recognized. The value of this attribute is the text that comprises the body of a named access list definition.
outacl#[1]	An output access list definition. For IP, standard or extended access list syntax can be used. For IPX, only extended syntax is recognized. The value of this attribute is the text that comprises the body of a named access list definition.
rte-fltr-in#	An input route filter. For IP, standard or extended access list syntax can be used, although you cannot mix them within a single list. For IPX, only extended syntax is recognized. The first line of this filter must specify a routing process. Subsequent lines comprise the body of a named access list.
rte-fltr-out#	An output route filter. For IP, standard or extended access list syntax can be used, although you cannot mix them within a single list. For IPX, only extended syntax is recognized. The first line of this filter must specify a routing process. Subsequent lines comprise the body of a named access list.
route#[2]	Static routes, for IP and IPX. The value is text of the form *destination-address mask* [*gateway*].
sap#	IPX static Service Advertising Protocol (SAP). The value is text from the body of an **ipx sap** configuration command.
sap-fltr-in#	IPX input SAP filter. Only extended access list syntax is recognized. The value is text from the body of an extended IPX **access-list** configuration command. (The Novell socket number for SAP filtering is 452.)
sap-fltr-out#	IPX output SAP filter. Only extended **access-list** command syntax is recognized. The value is text from the body of an extended IPX access-list configuration command.
pool-def#	An IP pool definition. The value is text from the body of an **ip local pool** configuration command.
pool-timeout	An IP pool definition. The body is an integer representing a timeout, in minutes.

1. The “outacl” attribute still exists and retains its old meaning.
2. The “route” attribute, without a trailing #, is still recognized for backward compatibility with the TACACS+ protocol specification, but if multiple static routes are required in TACACS+, full “route#” names will need to be employed.

Table 38 provides examples for each attribute on an AAA TACACS+ server.

Table 38 TACACS+ Server AV Pair Examples for Each Attribute

Attribute	TACACS+ Server Examples

inacl#	**IP:**
	`inacl#3="permit ip any any precedence immediate"`
	`inacl#4="deny igrp 10.0.1.2 255.255.0.0 any"`
	IPX:
	`inacl#1="deny 3C01.0000.0000.0001"`
	`inacl#2="deny 4C01.0000.0000.0002"`

outacl#	`outacl#2="permit ip any any precedence immediate"` `outacl#3="deny igrp 10.0.9.10 255.255.0.0 any"`

rte-fltr-in#	**IP:**
	`rte-fltr-in#1="router igrp 60"`
	`rte-fltr-in#3="permit 10.0.3.4 255.255.0.0"`
	`rte-fltr-in#4="deny any"`
	IPX:
	`rte-fltr-in#1="deny 3C01.0000.0000.0001"`
	`rte-fltr-in#2="deny 4C01.0000.0000.0002"`

rte-fltr-out#	`rte-fltr-out#1="router igrp 60"`
	`rte-fltr-out#3="permit 10.0.5.6 255.255.0.0"`
	`rte-fltr-out#4="permit any"`

route#	**IP:**
	`route#1="10.0.0.0 255.0.0.0 1.2.3.4"`
	`route#2="10.1.0.0 255.0.0.0"`
	IPX:
	`route#1="4C000000 ff000000 10.12.3.4"`
	`route#2="5C000000 ff000000 10.12.3.5"`

sap#	`sap#1="4 CE1-LAB 1234.0000.0000.0001 451 4"` `sap#2="5 CE3-LAB 2345.0000.0000.0001 452 5"`
sap-fltr-in#	`sap-fltr-in#1="deny 6C01.0000.0000.0001"` `sap-fltr-in#2="permit -1"`
sap-fltr-out#	`sap-fltr-out#1="deny 6C01.0000.0000.0001"` `sap-fltr-out#2="permit -1"`

pool-def#	`pool-def#1 = "aaa 10.0.0.1 1.0.0.3"`
	`pool-def#2 = "bbb 10.1.0.1 2.0.0.10"`
	`pool-def#3 = "ccc 10.2.0.1 3.0.0.20"`

pool-timeout	`pool-timeout=60`

Table 39 provides examples for each attribute on an AAA RADIUS server.

Table 39 RADIUS Server AV Pair Examples for Each Attribute

Attribute	RADIUS Server Examples
lcp:interface-config[1]	`cisco-avpair = "lcp:interface-config=ip address 10.0.0.0 255.255.255.0",`
inacl#	`cisco-avpair = "ip:inacl#3=permit ip any any precedence immediate",` `cisco-avpair = "ip:inacl#4=deny igrp 10.0.1.2 255.255.0.0 any",`

Table 39 RADIUS Server AV Pair Examples for Each Attribute (continued)

Attribute	RADIUS Server Examples
outacl#	`cisco-avpair = "ip:outacl#2=permit ip any any precedence immediate",` `cisco-avpair = "ip:outacl#3=deny igrp 10.0.9.10 255.255.0.0 any",`

rte-fltr-in#	**IP:**
	`cisco-avpair = "ip:rte-fltr-in#1=router igrp 60",`
	`cisco-avpair = "ip:rte-fltr-in#3=permit 10.0.3.4 255.255.0.0",`
	`cisco-avpair = "ip:rte-fltr-in#4=deny any",`
	IPX:
	`cisco-avpair = "ipx:rte-fltr-in=deny 3C01.0000.0000.0001",`

rte-fltr-out#	`cisco-avpair = "ip:rte-fltr-out#1=router igrp 60",`
	`cisco-avpair = "ip:rte-fltr-out#3=permit 10.0.5.6 255.255.0.0",`
	`cisco-avpair = "ip:rte-fltr-out#4=permit any",`

route#	**IP:**
	`cisco-avpair = "ip:route=3.10.0.0 255.0.0.0 1.2.3.4",`
	`cisco-avpair = "ip:route=4.10.0.0 255.0.0.0",`
	IPX:
	`cisco-avpair = "ipx:route=4C000000 ff000000 10.12.3.4",`
	`cisco-avpair = "ipx:route=5C000000 ff000000 10.12.3.5"`

sap#	`cisco-avpair = "ipx:sap=4 CE1-LAB 1234.0000.0000.0001 451 4",` `cisco-avpair = "ipx:sap=5 CE3-LAB 2345.0000.0000.0001 452 5",`
sap-fltr-in#	`cisco-avpair = "ipx:sap-fltr-in=deny 6C01.0000.0000.0001",` `cisco-avpair = "ipx:sap-fltr-in=permit -1"`
sap-fltr-out#	`cisco-avpair = "ipx:sap-fltr-out=deny 6C01.0000.0000.0001",` `cisco-avpair = "ipx:sap-fltr-out=permit -1"`

pool-def#	`cisco-avpair = "ip:pool-def#1=aaa 10.0.0.1 1.0.0.3",`
	`cisco-avpair = "ip:pool-def#2=bbb 10.1.0.1 2.0.0.10",`
	`cisco-avpair = "ip:pool-def#3=ccc 10.2.0.1 3.0.0.20",`

pool-timeout	`cisco-avpair = "ip:pool-timeout=60"`

1. This attribute is specific to RADIUS servers. It can be used to add Cisco IOS interface configuration commands to specific user configuration information.

How to Configure a AAA Server for Per-User Configuration

The configuration requirements and the structure of per-user configuration information is set by the specifications of each type of AAA server. Refer to your server documentation for more detailed information. The following sections about TACACS and RADIUS servers are specific to per-user configuration:

- Configuring a Freeware TACACS Server for Per-User Configuration (As required)
- Configuring a CiscoSecure TACACS Server for Per-User Configuration (As required)
- Configuring a RADIUS Server for Per-User Configuration (As required)

See the section "Monitoring and Debugging Per-User Configuration Settings" later in this chapter for tips on troubleshooting per-user configuration settings. See the section "Configuration Examples for Per-User Configuration" at the end of this chapter for examples of configuring RADIUS and TACACS servers.

Configuring a Freeware TACACS Server for Per-User Configuration

On a TACACS server, the entry in the user file takes a standard form. In the freeware version of TACACS+, the following lines appear in order:

- "User =" followed by the username, a space, and an open brace
- Authentication parameters
- Authorization parameters
- One or more AV pairs
- End brace on a line by itself

The general form of a freeware TACACS user entry is shown in the following example:

```
user = username {
     authentication parameters go here
     authorization parameters go here
}
```

The freeware TACACS user entry form is also shown by the following examples for specific users:

```
user= Router1
  Password= cleartext welcome
  Service= PPP protocol= ip {
        ip:route=10.0.0.0 255.0.0.0
        ip:route=10.1.0.0 255.0.0.0
        ip:route=10.2.0.0 255.0.0.0
        ip:inacl#5=deny 10.5.0.1
}

user= Router2
  Password= cleartext lab
  Service= PPP protocol= ip {
        ip:addr-pool=bbb
}
```

For more requirements and detailed information, refer to your AAA server documentation.

Configuring a CiscoSecure TACACS Server for Per-User Configuration

The format of an entry in the user file in the AAA database is generally name = value. Some values allow additional subparameters to be specified and, in these cases, the subparameters are enclosed in braces ({}). The following simple example depicts an AAA database showing the default user, one group, two users that belong to the group, and one user that does not:

```
# Sample AA Database 1
unknown_user = {
  password = system #Use the system's password file (/etc/passwd)
}
group = staff {
  # Password for staff who do not have their own.
  password = des "sefjkAlM7zybE"
  service = shell {
    # Allow any commands with any attributes.
    default cmd = permit
    default attribute = permit
  }
```

```
}
user = joe {  # joe uses the group password.

  member = "staff"
}
user = pete {  # pete has his own password.
  member = "staff"
  password = des "alkd9Ujiqp2y"
}
user = anita {
  # Use the "default" user password mechanism defined above.
  service = shell {
    cmd = telnet {  # Allow Telnet to any destination
    }
  }
}
```

For more information about the requirements and details of configuring the CiscoSecure server, see the *CiscoSecure UNIX Server User Guide*.

Configuring a RADIUS Server for Per-User Configuration

On a RADIUS server, the format of an entry in the users file includes the following lines in order:

- Username and password
- User service type
- Framed protocol
- One or more AV pairs

Note All these AV pairs are vendor specific. To use them, RADIUS servers must support the use of vendor-specific AV pairs. Patches for some servers are available from the Cisco Consulting Engineering (CE) customer-support organization.

The structure of an AV pair for Cisco platforms starts with *cisco-avpair* followed by a space, an equal sign, and another space. The rest of the line is within double quotation marks and, for all lines but the last, ends with a comma. Inside the double quotation marks is a phrase indicating the supported attribute, another equal sign, and a Cisco IOS command. The following examples show two different partial user configurations on a RADIUS server.

Router1

```
Password = "welcome"
User-Service-Type = Framed-User,
Framed-Protocol = PPP,
cisco-avpair = "ip:route=10.0.0.0 255.0.0.0",
cisco-avpair = "ip:route=10.1.0.0 255.0.0.0",
cisco-avpair = "ip:route=10.2.0.0 255.0.0.0",
cisco-avpair = "ip:inacl#5=deny 10.5.0.1"
```

Router2

```
Password = "lab"
User-Service-Type = Framed-User,
Framed-Protocol = PPP,
cisco-avpair = "ip:addr-pool=bbb"
```

Monitoring and Debugging Per-User Configuration Settings

Per-user configuration information exists on AAA servers only and is configured there, as described in the "How to Configure a AAA Server for Per-User Configuration" section.

For more information about configuring an application that can tie AAA per-user configuration information to generic interface and router configuration, see the chapter "Configuring Virtual Profiles" in this publication. Virtual profiles are required for combining per-user configuration information and generic interface and router configuration information to create virtual access interfaces for individual ISDN B channels.

However, you can monitor and debug the per-user configuration settings on the router or access server that are set from an AAA server. Table 40 indicates some of the commands to use for each attribute.

Table 40 Monitoring and Debugging Per-User Configuration Commands

Attribute	show Commands	debug Commands
inacl# outacl#	**show ip access-list** **show ip interface** *interface* **show ipx access-list** **show ipx interface**	**debug aaa authorization** **debug aaa per-user**
rte-fltr-in# rte-fltr-out#	**show ip access-list** **show ip protocols**	**debug aaa authorization** **debug aaa per-user**
route#	**show ip route** **show ipx route**	**debug aaa authorization** **debug aaa per-user**
sap#	**show ipx servers**	**debug aaa authorization** **debug aaa per-user**
sap-fltr-in# sap-fltr-out#	**show ipx access-list** **show ipx interface**	**debug aaa authorization** **debug aaa per-user**
pool-def# pool-timeout	**show ip local pool** [*name*]	—

Configuration Examples for Per-User Configuration

The following sections provide two comprehensive examples:

- TACACS+ Freeware Examples
- RADIUS Examples

These examples show router or access server configuration and AV pair configuration on an AAA server.

TACACS+ Freeware Examples

This section provides the TACACS+ freeware versions of the following examples:

- IP Access Lists and Static Routes Using Virtual Profiles over ISDN BRI
- IPX Per-User SAP Filters Using IPXWAN and Virtual Profiles by a Synchronous Interface

IP Access Lists and Static Routes Using Virtual Profiles over ISDN BRI

The following example provides configurations for the TACACS+ freeware daemon, the network access server, and the peer router named Router1. On the TACACS+ AAA server, peer router Router1 has a configuration that includes static routes and IP access lists.

TACACS+ Freeware Daemon Configuration File

```
key = tac123
user = Router1 {
global = cleartext welcome
service = ppp protocol = ip {
route#1=”10.0.0.0 255.0.0.0"
route#2=”10.1.0.0 255.0.0.0"
route#3=”10.2.0.0 255.0.0.0"
inacl#1=”deny 10.5.0.1"
}
}
```

Current Network Access Server Configuration

```
version 11.3
service timestamps debug datetime localtime
service udp-small-servers
service tcp-small-servers
!
hostname Router2
!
aaa new-model
aaa authentication ppp default tacacs+
aaa authorization network tacacs+
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
!
username Router1 password 7 15050E0007252621
ip host Router2 172.21.114.132
ip domain-name cisco.com
ip name-server 172.19.2.132
ip name-server 192.168.30.32
isdn switch-type basic-5ess
interface Ethernet 0
 ip address 172.21.114.132 255.255.255.224
 no ip mroute-cache
 media-type 10BaseT
!

interface Virtual-Template1
 ip unnumbered Ethernet0
 no cdp enable
!
!
interface BRI0
 ip unnumbered Ethernet0
 no ip mroute-cache
 encapsulation ppp
 no ip route-cache
 dialer idle-timeout 300
 dialer map ip 10.5.0.1 name Router1 broadcast 61482
 dialer-group 1
 no fair-queue
 ppp authentication chap
!
!
```

```
ip default-gateway 172.21.114.129
no ip classless
ip route 0.0.0.0 0.0.0.0 172.21.114.129
!
virtual-profile virtual-template 1
dialer-list 1 protocol ip permit
tacacs-server host 172.21.114.130
tacacs-server key tac123
```

Current Peer Configuration for Router1

```
version 11.3
no service pad
!
hostname Router1
!
enable secret 5 $1$m1WK$RsjborN1Z.XZuFqsrtSnp/
enable password lab
!
username Router2 password 7 051C03032243430C
ip host Router1 172.21.114.134
ip domain-name cisco.com
ip name-server 172.19.2.132
ip name-server 192.168.30.32
isdn switch-type basic-5ess
!
interface Ethernet0
 ip address 172.21.114.134 255.255.255.224
 no ip route-cache
 shutdown
!
interface BRI0
 ip address 10.5.0.1 255.0.0.0
 encapsulation ppp
 dialer map ip 172.21.114.132 name Router2 broadcast 61483
 dialer-group 1
 no fair-queue
!
ip default-gateway 172.21.114.129
no ip classless
ip route 172.21.0.0 255.255.0.0 BRI0
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password lab
 login
end
```

IPX Per-User SAP Filters Using IPXWAN and Virtual Profiles by a Synchronous Interface

The following example provides configurations for the TACACS+ daemon and the peer router named Router1. On the TACACS+ AAA server, user ny has a configuration that includes inbound and outbound SAP filters.

TACACS+ Freeware Daemon Configuration File for User

```
key = tac123
user = Router1 {
 global = cleartext welcome
 service = ppp protocol = ipx {
        sap=”101 CYBER-01 40.0000.0000.0001 400 10"
        sap=”202 CYBER-02 40.0000.0000.0001 401 10"
        sap=”303 CYBER-03 40.0000.0000.0001 402 10"
        sap-fltr-out#1=”deny 40 101"
        sap-fltr-out#2=”deny 40 202"
        sap-fltr-out#3=”permit -1"
        sap-fltr-in#1=”permit 30 444"
        sap-fltr-in#2=”deny -1"
```

Current Remote Peer (Router1) Configuration

```
version 11.3
!
hostname Router1
!
enable password lab
!
username Router2 password 7 140017070F0B272E
ip host Router1 172.21.114.131
ip name-server 172.19.2.132
ip name-server 192.168.30.32
ipx routing 0000.0c47.090d
ipx internal-network 30
!
interface Ethernet0
 ip address 172.21.114.131 255.255.255.224
!
interface Serial1
 no ip address
 encapsulation ppp
 ipx ipxwan 0 unnumbered peer-Router1
 clockrate 4000000
!
ipx sap 444 ZEON-4 30.0000.0000.0001 444 10
ipx sap 555 ZEON-5 30.0000.0000.0001 555 10
ipx sap 666 ZEON-6 30.0000.0000.0001 666 10
!
Current Network Access Server (Router2) Configuration
version 11.3
service timestamps debug uptime
!
hostname Router2
!
aaa new-model
aaa authentication ppp default tacacs+
aaa authorization network tacacs+
enable password lab
!
username Router1 password 7 044C0E0A0C2E414B
ip host LA 172.21.114.133
ip name-server 192.168.30.32
```

```
ip name-server 172.19.2.132
ipx routing 0000.0c47.12d3
ipx internal-network 40
!
interface Ethernet0
 ip address 172.21.114.133 255.255.255.224
!
interface Virtual-Template1
 no ip address
 ipx ipxwan 0 unnumbered nas-Router2
 no cdp enable
!
interface Serial1
 ip unnumbered Ethernet0
 encapsulation ppp
 ipx ipxwan 0 unnumbered nas-Router2
 ppp authentication chap
!
ipx sap 333 DEEP9 40.0000.0000.0001 999 10
!
virtual-profile virtual-template 1
tacacs-server host 172.21.114.130
tacacs-server key tac123
```

RADIUS Examples

This section provides the RADIUS versions of the following examples:

- IP Access Lists and Static Routes Using Virtual Profiles over ISDN BRI
- IPX Per-User SAP Filters Using IPXWAN and Virtual Profiles by a Synchronous Interface

IP Access Lists and Static Routes Using Virtual Profiles over ISDN BRI

The following example shows a remote peer (Router1) configured to dial in to a BRI on a Cisco network access server (Router2), which requests user configuration information from an AAA server (radiusd):

RADIUS User File (Router1)

```
Password = "welcome"
User-Service-Type = Framed-User,
Framed-Protocol = PPP,
cisco-avpair = "ip:route=10.1.0.0 255.0.0.0",
cisco-avpair = "ip:route=10.2.0.0 255.0.0.0",
cisco-avpair = "ip:route=10.3.0.0 255.0.0.0",
cisco-avpair = "ip:inacl#5=deny 10.0.0.1"
```

Current Network Access Server Configuration

```
version 11.3
service timestamps debug datetime localtime
service udp-small-servers
service tcp-small-servers
!
hostname Router2
!
aaa new-model
aaa authentication ppp default radius
aaa authorization network radius
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
```

```
!
username Router1 password 7 15050E0007252621
ip host Router2 172.21.114.132
ip domain-name cisco.com
ip name-server 172.19.2.132
ip name-server 192.168.30.32
isdn switch-type basic-5ess
interface Ethernet0
 ip address 172.21.114.132 255.255.255.224
 no ip mroute-cache
 media-type 10BaseT
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 no cdp enable
!
interface BRI0
 ip unnumbered Ethernet0
 no ip mroute-cache
 encapsulation ppp
 no ip route-cache
 dialer idle-timeout 300
 dialer map ip 10.5.0.1 name Router1 broadcast 61482
 dialer-group 1
 no fair-queue
 ppp authentication chap
!
ip default-gateway 172.21.114.129
no ip classless
ip route 0.0.0.0 0.0.0.0 172.21.114.129
!
virtual-profile vtemplate 1
dialer-list 1 protocol ip permit
radius-server host 172.21.114.130
radius-server key rad123
```

Current Peer Configuration for Router1

```
version 11.3
no service pad
!
hostname Router1
!
enable secret 5 $1$m1WK$RsjborN1Z.XZuFqsrtSnp/
enable password lab
!
username Router2 password 7 051C03032243430C
ip host Router1 172.21.114.134
ip domain-name cisco.com
ip name-server 172.19.2.132
ip name-server 192.168.30.32
isdn switch-type basic-5ess
!
interface Ethernet0
 ip address 172.21.114.134 255.255.255.224
 no ip route-cache
 shutdown
!
interface BRI0
 ip address 10.5.0.1 255.0.0.0
 encapsulation ppp
 dialer map ip 172.21.114.132 name Router2 broadcast 61483
 dialer-group 1
 no fair-queue
```

```
!
ip default-gateway 172.21.114.129
no ip classless
ip route 172.21.0.0 255.255.0.0 BRI0
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password lab
 login
!
end
```

Output of ping Command from Router1

```
Router1# ping 172.21.114.132

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.21.114.132, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

(fails due to access list deny)
```

RADIUS Debug Output

```
radrecv: Request from host ac157284 code=1, id=46, length=67
    Client-Id = 172.21.114.132
    Client-Port-Id = 1112670208
    User-Name = "Router1"
    CHAP-Password = "\037\317\213\326*\236)#+\266\243\255x\331\370v\334"
    User-Service-Type = Framed-User
    Framed-Protocol = PPP
Sending Ack of id 46 to ac157284 (172.21.114.132)
    User-Service-Type = Framed-User
    Framed-Protocol = PPP
    [Vendor 9] cisco-avpair = "ip:route=10.0.0.0 255.0.0.0"
    [Vendor 9] cisco-avpair = "ip:route=10.1.0.0 255.0.0.0"
    [Vendor 9] cisco-avpair = "ip:route=10.2.0.0 255.0.0.0"
    [Vendor 9] cisco-avpair = "ip:inacl#5=deny 10.0.0.1"
```

Network Access Server (Router2) show and debug Command Output

```
Router2# show debug

General OS:
  AAA Authorization debugging is on
PPP:
  PPP authentication debugging is on
  Multilink activity debugging is on
ISDN:
  ISDN events debugging is on
Dial on demand:
  Dial on demand events debugging is on
VTEMPLATE:
  Virtual Template debugging is on

pr  4 08:30:09: ISDN BR0: received HOST_INCOMING_CALL
        Bearer Capability i = 0x080010
*Apr  4 08:30:09:       --------------------
        Channel ID i = 0x0101
*Apr  4 08:30:09:         IE out of order or end of 'private' IEs --
        Bearer Capability i = 0x8890
```

```
*Apr  4 08:30:09:         Channel ID i = 0x89
*Apr  4 08:30:09:         Called Party Number i = 0xC1, '61483'
*Apr  4 08:30:09: ISDN BR0: Event: Received a call from <unknown> on B1 at 64 Kb/s
*Apr  4 08:30:09: ISDN BR0: Event: Accepting the call
%LINK-3-UPDOWN: Interface BRI0:1, changed state to up
*Apr  4 08:30:09: ISDN BR0: received HOST_CONNECT
       Channel ID i = 0x0101
*Apr  4 08:30:09:       -------------------
       Channel ID i = 0x89
*Apr  4 08:30:09: ISDN BR0: Event: Connected to <unknown> on B1 at 64 Kb/s
*Apr  4 08:30:09: PPP BRI0:1: Send CHAP challenge id=30 to remote
*Apr  4 08:30:10: PPP BRI0:1: CHAP response received from Router1
*Apr  4 08:30:10: PPP BRI0:1: CHAP response id=30 received from Router1
*Apr  4 08:30:10: AAA/AUTHOR/LCP: authorize LCP
*Apr  4 08:30:10: AAA/AUTHOR/LCP: BRI0:1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/LCP: BRI0:1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/LCP: BRI0:1: (0): send AV protocol=lcp
*Apr  4 08:30:10: AAA/AUTHOR/LCP: BRI0:1: (2084553184): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (2084553184): Post authorization status = PASS_ADD
*Apr  4 08:30:10: PPP BRI0:1: Send CHAP success id=30 to remote
*Apr  4 08:30:10: PPP BRI0:1: remote passed CHAP authentication.
*Apr  4 08:30:10: VTEMPLATE Reuse vaccess1, New Recycle queue size:0

*Apr  4 08:30:10: VTEMPLATE set default vaccess1 with no ip address

*Apr  4 08:30:10: Virtual-Access1 VTEMPLATE hardware address 0000.0c46.154a
*Apr  4 08:30:10: VTEMPLATE vaccess1 has a new cloneblk vtemplate, now it has vtemplate
*Apr  4 08:30:10: VTEMPLATE undo default settings vaccess1

*Apr  4 08:30:10: VTEMPLATE ************* CLONE VACCESS1 ******************Apr  4
08:30:10: VTEMPLATE Clone from vtemplate1 to vaccess1
interface Virtual-Access1
no ip address
encap ppp
ip unnumbered ethernet 0
end

%LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
*Apr  4 08:30:10: AAA/AUTHOR/LCP: authorize LCP
*Apr  4 08:30:10: AAA/AUTHOR/LCP: Virtual-Access1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/LCP: Virtual-Access1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/LCP: Virtual-Access1: (0): send AV protocol=lcp
*Apr  4 08:30:10: AAA/AUTHOR/LCP: Virtual-Access1: (1338953760): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (1338953760): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): can we start IPCP?
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): send AV protocol=ip
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (1716082074): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (1716082074): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: we can start IPCP (0x8021)
*Apr  4 08:30:10: MLP Bad link Virtual-Access1
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): can we start UNKNOWN?
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): send AV protocol=unknown
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (2526612868): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (2526612868): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: we can start UNKNOWN (0x8207)
*Apr  4 08:30:10: MLP Bad link Virtual-Access1
*Apr  4 08:30:10: BRI0:1: Vaccess started from dialer_remote_name
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): can we start IPCP?
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): send AV service=ppp
```

```
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): send AV protocol=ip
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (3920403585): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (3920403585): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: we can start IPCP (0x8021)
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): can we start UNKNOWN?
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): send AV protocol=unknown
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (3439943223): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (3439943223): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: we can start UNKNOWN (0x8207)
%LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: start: her address 10.0.0.1, we want
0.0.0.0
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (0): user='Router1'
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (0): send AV servi*Apr  4 08:30:13:
AAA/AUTHOR/IPCP: Virtual-Access1: (0): send AV service=ppp
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (0): send AV protocol=ip
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (0): send AV addr*10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (3215797579): Method=RADIUS
*Apr  4 08:30:13: AAA/AUTHOR (3215797579): Post authorization status = PASS_ADD
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV service=ppp
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV protocol=ip
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV addr*10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV route=10.1.0.0 255.0.0.0
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV route=10.2.0.0 255.0.0.0
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV route=10.3.0.0 255.0.0.0
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV inacl#5=deny 10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: authorization succeeded
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: done: her address 10.0.0.1, we want
10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: authorization succeeded
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: parse_cmd 'ip route 10.0.0.0 255.0.0.0
10.0.0.1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: enqueue peruser IP txt=no ip route 10.0.0.0
255.0.0.0 10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: parse_cmd 'ip route 11.0.0.0 255.0.0.0
10.0.0.1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: enqueue peruser IP txt=no ip route 11.0.0.0
255.0.0.0 10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: parse_cmd 'ip route 12.0.0.0 255.0.0.0
10.0.0.1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: enqueue peruser IP txt=no ip route 12.0.0.0
255.0.0.0 10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR: parse 'ip access-list standard Virtual-Access1#1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: parse 'deny 10.0.0.1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: enqueue peruser IP txt=no ip access-list
standard Virtual-Access1#1
*Apr  4 08:30:13: VTEMPLATE vaccess1 has a new cloneblk AAA, now it has vtemplate/AAA
*Apr  4 08:30:13: VTEMPLATE ************* CLONE VACCESS1 *****************

*Apr  4 08:30:13: VTEMPLATE Clone from AAA to vaccess1
interface Virtual-Access1
ip access-group Virtual-Access1#1 in

*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: vaccess parse 'interface Virtual-Access1
ip access-group Virtual-Access1#1 in
' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR/FSM: Check for unauthorized mandatory AV's
*Apr  4 08:30:13: AAA/AUTHOR/FSM: Processing AV service=ppp
*Apr  4 08:30:13: AAA/AUTHOR/FSM: Processing AV protocol=unknown
*Apr  4 08:30:13: AAA/AUTHOR/FSM: succeeded
%ISDN-6-CONNECT: Interface BRI0:1 is now connected to Router1
```

```
Router2# show ip access-list

Standard IP access list Virtual-Access1#1 (per-user)
    deny   10.0.0.1

Router2# show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is 172.21.114.129 to network 0.0.0.0

U    10.0.0.0/8 [1/0] via 10.3.0.1
U    10.1.0.0/8 [1/0] via 10.3.0.1
U    10.2.0.0/8 [1/0] via 10.3.0.1
     10.3.0.0/8 is subnetted, 1 subnets
C       10.3.0.1 is directly connected, Virtual-Access1
     172.21.0.0/16 is subnetted, 1 subnets
C       172.21.114.128 is directly connected, Ethernet0
S*   0.0.0.0/0 [1/0] via 172.21.114.129


Router2# show interfaces virtual-access 1

Virtual-Access1 is up, line protocol is up
  Hardware is Virtual Access interface
  Interface is unnumbered.  Using address of Ethernet0 (172.21.114.132)
  MTU 1500 bytes, BW 64 Kbit, DLY 100000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive set (10 sec)
  DTR is pulsed for 5 seconds on reset
  LCP Open, multilink Closed
  Open: IPCP, CDP
  Last input 5d04h, output never, output hang never
  Last clearing of "show interface" counters 00:06:42
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     76 packets input, 3658 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     141 packets output, 2909 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions

Router2# show ip interface virtual-access 1

Virtual-Access1 is up, line protocol is up
  Interface is unnumbered.  Using address of Ethernet0 (172.21.114.132)
  Broadcast address is 255.255.255.255
  Peer address is 10.0.0.1
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is enabled
  Outgoing access list is not set
  Inbound  access list is Virtual-Access1#1
  Proxy ARP is enabled
  Security level is default
```

```
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is disabled


Router2# debug ip packet

IP packet debugging is on
Router2#
*Apr  4 08:30:42: IP: s=172.21.114.129 (Ethernet0), d=255.255.255.255, len 186, rcvd 2
*Apr  4 08:30:42: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, a*Apr  4
08:30:42: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, access denied
*Apr  4 08:30:42: IP: s=172.21.114.132 (local), d=10.0.0.1 (Virtual-Access1), len 4,
sending
*Apr  4 08:30:42: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, access
denied
*Apr  4 08:30:44: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, access
denied
*Apr  4 08:30:44: IP: s=172.21.114.132 (local), d=10.0.0.1 (Virtual-Access1), len 16,
sending
*Apr  4 08:30:44: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, access
denied
```

IPX Per-User SAP Filters Using IPXWAN and Virtual Profiles by a Synchronous Interface

The following examples show a remote peer (Router1) configured to dial in to a synchronous interface on a Cisco network access server (Router2), which requests user configuration information from an AAA server (radiusd):

RADIUS User File (Router 1)

```
Password = "welcome"
User-Service-Type = Framed-User,
Framed-Protocol = PPP,
cisco-avpair = "ipx:sap=101 CYBER-01 40.0000.0000.0001 400 10",
cisco-avpair = "ipx:sap=202 CYBER-02 40.0000.0000.0001 401 10",
cisco-avpair = "ipx:sap=303 CYBER-03 40.0000.0000.0001 402 10",
cisco-avpair = "ipx:sap-fltr-out#20=deny 40 101",
cisco-avpair = "ipx:sap-fltr-out#21=deny 40 202",
cisco-avpair = "ipx:sap-fltr-out#22=permit -1",
cisco-avpair = "ipx:sap-fltr-in#23=permit 30 444",
cisco-avpair = "ipx:sap-fltr-in#23=deny -1"
```

Current Remote Peer (Router 1) Configuration

```
hostname Router1
!
enable password lab
!
username Router2 password 7 140017070F0B272E
ip host Router1 172.21.114.131
ip name-server 172.19.2.132
ip name-server 192.168.30.32
ipx routing 0000.0c47.090d
ipx internal-network 30
!
interface Ethernet0
 ip address 172.21.114.131 255.255.255.224
!
```

```
interface Serial1
 no ip address
 encapsulation ppp
 ipx ipxwan 0 unnumbered peer-Router1
 clockrate 4000000
!
ipx sap 444 ZEON-4 30.0000.0000.0001 444 10
ipx sap 555 ZEON-5 30.0000.0000.0001 555 10
ipx sap 666 ZEON-6 30.0000.0000.0001 666 10
!
...
version 12.1
service timestamps debug uptime
!
hostname Router2
!
aaa new-model
aaa authentication ppp default radius
aaa authorization network radius
enable password lab
!
username Router1 password 7 044C0E0A0C2E414B
ip host Router2 172.21.114.133
ip name-server 172.22.30.32
ip name-server 192.168.2.132
ipx routing 0000.0c47.12d3
ipx internal-network 40
!
interface Ethernet0
 ip address 172.21.114.133 255.255.255.224
!
interface Virtual-Template1
 no ip address
 ipx ipxwan 0 unnumbered nas-Router2
 no cdp enable
!
interface Serial1
 ip unnumbered Ethernet0
 encapsulation ppp
 ipx ipxwan 0 unnumbered nas-Router2
 ppp authentication chap
!
ipx sap 333 DEEP9 40.0000.0000.0001 999 10
!
virtual-profile vtemplate 1
radius-server host 172.21.114.130
radius-server key rad123
```

RADIUS debug Output

```
radrecv: Request from host ac157285 code=1, id=23, length=67
    Client-Id = 172.21.114.133
    Client-Port-Id = 1399128065
    User-Name = "Router1"
    CHAP-Password = "%"(\012I$\262\352\031\276\024\302\277\225\347z\274"
    User-Service-Type = Framed-User
    Framed-Protocol = PPP
Sending Ack of id 23 to ac157285 (172.21.114.133)
    User-Service-Type = Framed-User
    Framed-Protocol = PPP
    [Vendor 9] cisco-avpair = "ipx:sap=101 CYBER-01 40.0000.0000.0001 400 10"
    [Vendor 9] cisco-avpair = "ipx:sap=202 CYBER-02 40.0000.0000.0001 401 10"
    [Vendor 9] cisco-avpair = "ipx:sap=303 CYBER-03 40.0000.0000.0001 402 10"
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-out#20=deny1 40 101"
```

```
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-out#21=deny 40 202"
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-out#22=permit -1"
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-in#23=permit 30 444"
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-in#23=deny -1"
```

Network Access Server show Command Output

```
Router2# show ipx servers

Codes: S - Static, P - Periodic, E - EIGRP, N - NLSP, H - Holddown, + = detail
5 Total IPX Servers

Table ordering is based on routing and server info

   Type Name                          Net     Address    Port     Route Hops Itf
s   101 CYBER-01                      40.0000.0000.0001:0400      conn  10   Int
s   202 CYBER-02                      40.0000.0000.0001:0401      conn  10   Int
s   303 CYBER-03                      40.0000.0000.0001:0402      conn  10   Int
S   333 DEEP9                         40.0000.0000.0001:0999      conn  10   Int
P   444 ZEON-4                        30.0000.0000.0001:0444      7/01  11   Vi1


Router1# show ipx servers

Codes: S - Static, P - Periodic, E - EIGRP, N - NLSP, H - Holddown, + = detail
5 Total IPX Servers

Table ordering is based on routing and server info

   Type Name                          Net     Address    Port     Route Hops Itf
P   303 CYBER-03                      40.0000.0000.0001:0402      7/01  11   Se1
P   333 DEEP9                         40.0000.0000.0001:0999      7/01  11   Se1
S   444 ZEON-4                        30.0000.0000.0001:0444      conn  10   Int
S   555 ZEON-5                        30.0000.0000.0001:0555      conn  10   Int
S   666 ZEON-6                        30.0000.0000.0001:0666      conn  10   Int

Router2# show ipx access-list

IPX sap access list Virtual-Access1#2
    permit 30 444
    deny FFFFFFFF
IPX sap access list Virtual-Access1#3
    deny 40 101
    deny 40 202
    permit FFFFFFFF
```

Configuring Resource Pool Management

This chapter describes the Cisco Resource Pool Management (RPM) feature. It includes the following main sections:

- RPM Overview
- How to Configure RPM
- Verifying RPM Components
- Troubleshooting RPM
- Configuration Examples for RPM

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature, or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

For a complete description of the commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

RPM Overview

Cisco RPM enables telephone companies and Internet service providers (ISPs) to share dial resources for wholesale and retail dial network services. With RPM, telcos and ISPs can count, control, and manage dial resources and provide accounting for shared resources when implementing different service-level agreements.

You can configure RPM in a single, standalone Cisco network access server (NAS) by using RPM or, optionally, across multiple NAS stacks by using one or more external Cisco Resource Pool Manager Servers (RPMS).

Cisco RPM gives data network service providers the capability to do the following:

- Have the flexibility to include local retail dial services in the same NAS with the wholesale dial customers.
- Manage customer use of shared resources such as modems or High-Level Data Link Control (HDLC) controllers for data calls.
- Offer advanced wholesale dialup services using a Virtual Private Dialup Network (VPDN) to enterprise accounts and ISPs.
- Deploy Data over Voice Bearer Service (DoVBS).

- Manage call sessions by differentiating dial customers through customer profiles. The customer profile determines where resources are allocated and is based on the incoming Dialed Number Information Service (DNIS) number or Calling Line Identification (CLID).
- Efficiently use resource groups such as modems to offer differing over subscription rates and dial service-level agreements.

Note Ear and Mouth Feature Group B (E&M-FGB) is the only signaling type supported for channel-associated signaling (CAS) on T1 and T3 facilities; R2 is supported for E1 facilities. FG D is not supported. Cisco IOS software collects DNIS digits for the signaling types FGB, PRI, and SS7 and only E&M-FGB and R2 CAS customer profiles are supported. For all other CAS signaling types, use the default DNIS group customer profiles.

Components of Incoming and Outgoing Call Management

Cisco RPM manages both incoming calls and outgoing sessions. Cisco RPM differentiates dial customers through configured customer profiles based on the DNIS and call type determined at the time of an incoming call.

The components of incoming call management in the Cisco RPM are described in the following sections:

- Customer Profile Types
- DNIS Groups
- Call Types
- Resource Groups
- Resource Services

You can use Cisco RPM to answer all calls and differentiate customers by using VPDN profiles and groups. The components of outgoing session management in the Cisco RPM are described in the following sections:

- VPDN Groups
- VPDN Profiles

Note These components of Cisco RPM are enabled after the NAS and other equipment has been initially set up, configured, and verified for proper operation of the dial, PPP, VPDN, and authentication, authorization, and accounting (AAA) segments. Refer to the Cisco IOS documentation for these other segments for installation, configuration, and troubleshooting information before attempting to use RPM.

Configured DNIS groups and resource data can be associated to customer profiles. These customer profiles are selected by the incoming call DNIS number and call type and then used to identify resource allocations based on the associated resource groups and defined resource services.

After the call is answered, customer profiles can also be associated with VPDN groups so the configured VPDN sessions and other data necessary to set up or reject a VPDN session are applied to the answered calls. VPDN group data includes associated domain name or DNIS, IP addresses of endpoints, maximum sessions per endpoint, maximum Multilink PPP (MLP) bundles per VPDN group, maximum links per MLP bundle, and other tunnel information.

Customer Profile Types

There are three types of customer profiles in Cisco RPM, which are described in the following sections:

- Customer Profiles
- Default Customer Profiles
- Backup Customer Profiles

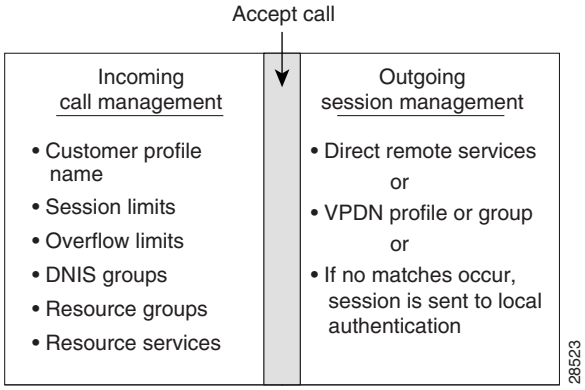
Additionally, you can create a customer profile template and associate it with a customer profile; it is then integrated into the customer profile.

Customer Profiles

A customer profile defines how and when to answer a call. Customer profiles include the following components (see Figure 99):

- Customer profile name and description—Name and description of the customer.
- Session limits—Maximum number of standard sessions.
- Overflow limits—Maximum number of overflow sessions.
- DNIS groups.
- CLID.
- Resource groups.
- Resource services.
- VPDN groups and VPDN profiles.
- Call treatment—Determines how calls that exceed the session and overflow limits are treated.

Figure 99 Components of a Customer Profile



The incoming side of the customer profile determines if the call will be answered using parameters such as DNIS and call type from the assigned DNIS group and session limits. The call is then assigned the appropriate resource within the resource group defined in the customer profile. Each configured customer profile includes a maximum allowed session value and an overflow value. As sessions are started and ended, session counters are incremented and decremented so customer status is kept current. This information is used to monitor the customer resource limit and determine the appropriate call treatment based on the configured session limits.

The outgoing side of the customer profile directs the answered call to the appropriate destination:

- To a local AAA server of retail dial applications and Internet/intranet access.
- To a tunnel that is established between the NAS or L2TP Access Concentrator (LAC) to a wholesale VPDN home gateway of a dial customer, or L2TP Network Server (LNS) using Layer 2 Forwarding Protocol (L2F) or Layer 2 Tunneling Protocol (L2TP) technology.

Default Customer Profiles

Default customer profiles are identical to standard customer profiles, except that they do not have any associated DNIS groups. Default customer profiles are created using the reserved keyword **default** for the DNIS group.

Default customer profiles are used to provide session counting and resource assignment to incoming calls that do not match any of the configured DNIS groups. Although specific resources and DNIS groups can be assigned to customer profiles, default customer profiles allow resource pooling for the calls that do not match the configured DNIS groups or where the DNIS is not provided. Retail dial services and domain-based VPDN use default customer profiles.

When multiple default customer profiles are used, the call type (speech, digital, V.110, or V.120) of the default DNIS group is used to identify which default customer profile to use for an incoming call. At most, four default profiles (one for each call type) can be configured.

Note If default customer profiles are not defined, then calls that do not match a DNIS group in a customer profile are rejected with a "no answer" or "busy" call treatment sent to the switch.

Backup Customer Profiles

Backup customer profiles are customer profiles configured locally on the Cisco NAS and are used to answer calls based on a configured allocation scheme when the link between the Cisco NAS and Cisco RPMS is disabled. See the section "Configuring Customer Profiles Using Backup Customer Profiles" for more information about configuring backup customer profiles.

Customer Profile Template

With RPM, users can also implement wholesale dial services without using VPDN tunnels to complete dial-in calls to destinations of the end customer. This capability is accomplished with components of the AAA groups and the PPP configurations.

The AAA group provides IP addresses of AAA servers for authentication and accounting. The PPP configurations allow users to configure the Cisco IOS PPP feature set on each customer profile. In this current implementation, PPP configuration is based on the following:

- Applicable IP address pool(s) or default local list of IP addresses
- Primary and secondary Domain Name System (DNS) or Windows Internet naming service (WINS)
- Number of links allowed for each call using MLP

Note The AAA and PPP integration applies to a single NAS environment.

To add PPP configurations to a customer profile, you must create a customer profile template. Once you create the template and associate it with a customer profile using the **source template** command, it is integrated into the customer profile.

The RPM customer profile template for the PPP command set, when used with the Cisco IOS feature, Server Groups Selected by DNIS, presents a strong single NAS solution for providers of wholesale dial services, as follows:

- Call acceptance is determined by the RPM before call answering, using the configured size limits and resource availability.
- The answered call then uses the PPP configuration defined in the template to initiate authentication, obtain an IP address, and select a DNS or WINS that is located at the customer site.
- The same DNIS that was used to choose the customer profile selects the servers for authentication/authorization and accounting that are located at the wholesale customer's site.

The section "Configuring a Customer Profile Template" later in this chapter describes how to create a customer profile template so that you can configure the Cisco IOS PPP features on a customer profile, but this section does not list the existing PPP command set. For information about the PPP command set, refer to the *Cisco IOS Dial Technologies Command Reference*.

DNIS Groups

A DNIS group is a configured list of DNIS called party numbers that correspond to the numbers dialed to access particular customers, service offerings, or both. For example, if a customer from phone number 000-1234 calls a number 000-5678, the DNIS provides information on the number dialed—000-5678.

Cisco RPM checks the DNIS number of inbound calls against the configured DNIS groups, as follows:

- If Cisco RPM finds a match, it uses the configured information in the customer profile to which the DNIS group is assigned.
- If Cisco RPM does not find a match, it uses the configured information in the customer profile to which the default DNIS group is assigned.
- The DNIS/call type sequence can be associated only with one customer profile.

CLID Groups

A CLID group is a configured list of CLID calling party numbers. The CLID group specifies a list of numbers to reject if the group is associated with a call discriminator. For example, if a customer from phone number 000-1234 calls a number 000-5678, the CLID provides information on the calling party number—000-1234.

A CLID can be associated with only one CLID group.

Call Types

Call types from calls originating from ISDN, SS7, and CAS (CT1, CT3, and CE1) are used to assign calls to the appropriate resource. Call types for ISDN and SS7 are based on Q.931 bearer capability. Call types for CAS are assigned based on static channel configuration.

Supported call types are as follows:

- Speech
- Digital
- V.110
- V.120

Note Voice over IP, fax over IP, and dial-out calls are not supported in RPM.

Resource Groups

Cisco RPM enables you to maximize the use of available shared resources within a Cisco NAS for various resource allocation schemes to support service-level agreements. Cisco RPM allows you to combine your Cisco NAS resource groups with call types (speech, digital, V.110, and V.120) and optional resource modem services. Resource groups and services are configured for customer profiles and assigned to incoming calls through DNIS groups and call types.

Resource groups have the following characteristics:

- Are configured on the Cisco NAS and applied to a customer profile.
- Represent groupings of similar hardware or firmware that are static and do not change on a per-call basis.
- Can define resources that are port-based or not port-based:
  - Port-based resources are identified by physical location, such as a range of port/slot numbers (for example, modems or terminal adapters).
  - Non-port-based resources are identified by a single size parameter (for example, HDLC framers or V.120 terminal adapters—V.120 terminal adapters are currently implemented as part of Cisco IOS software).

Resource assignments contain combinations of Cisco NAS resource groups, optional resource modem services, and call types. The NAS resources in resource groups that have not been assigned to a customer profile will not be used.

Note To support ISDN DoVBS, use a DNIS group and a configured customer profile to direct the speech call to the appropriate digital resource. The resource group assigned to this customer profile will be "digital resources" and also have a call type of "speech," so the call will terminate on an HDLC controller rather than a modem.

Resource Services

A resource service contains a finite series of resource command strings that can be used to help dynamically configure an incoming connection. Services supported by a resource group are determined by the combination of hardware and firmware installed. Currently, resource service options can be configured and applied to resource groups. Resource services can be defined to affect minimum and maximum speed, modulation, error correction, and compression, as shown in Table 41.

Table 41 Resource Services

Service	Options	Comments
min-speed	<300–56000>, any	Must be a V.90 increment.
max-speed	<300–56000>, any	Must be a V.90 increment.
modulation	k56flex, v22bis, v32bis, v34, v90, any	None.

Table 41 Resource Services (continued)

Service	Options	Comments
error-correction	1apm, mn14	This is a hidden command.
compression	mnps, v42bis	This is a hidden command.

VPDN Groups

The VPDN group contains the data required to build a VPDN tunnel from the RPM NAS LAC to the LNS. In the context of RPM, VPDN is authorized by first associating a customer profile with a VPDN group, and second by associating the VPDN group to the DNIS group used for that customer profile. VPDN group data includes the endpoint IP addresses.

Cisco RPM enables you to specify multiple IP endpoints for a VPDN group, as follows:

- If two or more IP endpoints are specified, Cisco RPM uses a load-balancing method to ensure that traffic is distributed across the IP endpoints.
- For DNIS-based VPDN dial service, VPDN groups are assigned to customer profiles based on the incoming DNIS number and the configured DNIS groups.
- For domain-based VPDN dial service, VPDN groups are assigned to the customer profile or the default customer profile with the matching call-type assignment.
- For either DNIS-based or domain-based VPDN dial services, there is a customer profile or default customer profile for the initial resource allocation and customer session limits.

The VPDN group provides call management by allowing limits to be applied to both the number of MLP bundles per tunnel and the number of links per MLP bundle. Limits can also restrict the number of sessions per IP endpoint. If you require more granular control of VPDN counters, use VPDN profiles.

VPDN Profiles

VPDN profiles allow session and overflow limits to be imposed for a particular customer profile. These limits are unrelated to the limits imposed by the customer profile. A customer profile is associated with a VPDN profile. A VPDN profile is associated with a VPDN group. VPDN profiles are required only when these additional counters are required for VPDN usage per customer profile.

Call Treatments

Call treatment determines how calls are handled when certain events require the call to be rejected. For example, if the session and overflow limits for one of your customers have been exceeded, any additional calls will receive a busy signal (see Table 42).

Table 42 Call-Treatment Table

Event	Call-Treatment Option	Results
Customer profile not found	No answer (default)	The caller receives rings until the switch eventually times out. Implies that the NAS was appropriate, but resources were unavailable. The caller should try later.
	Busy	The switch drops the call from the NAS and sends a busy signal back to the caller. The call is rejected based on not matching a DNIS group/call type and customer profile. Can be used to immediately reject the call and free up the circuit.
Customer profile limits exceeded	Busy	The switch drops the call from the NAS and sends a busy signal back to the caller.
NAS resource not available	Channel not available (default)	The switch sends the call to the next channel in the trunk group. The call can be answered, but the NAS does not have any available resources in the resource groups. Allows the switch to try additional channels until it gets to a different NAS in the same trunk group that has the available resources.
	Busy	The switch drops the call from the NAS and sends a busy signal back to the caller. Can be used when the trunk group does not span additional NASes.
Call discrimination match	No answer	The caller receives rings until the switch eventually times out.

Details on RPM Call Processes

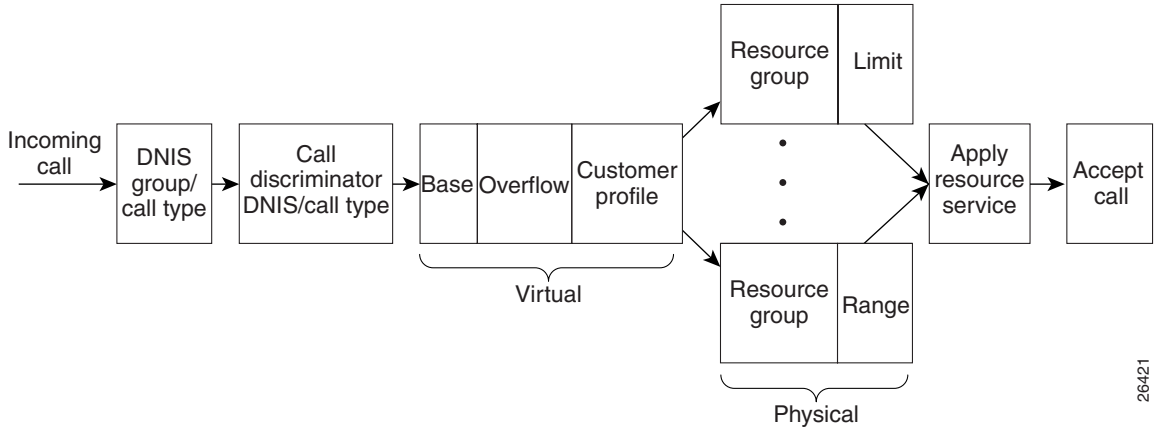
On the incoming call management of the customer profile, the following sequence occurs to determine if a call is answered:

1. The incoming DNIS is mapped to a DNIS group; if there is no incoming DNIS number, or the DNIS number provided does not match any configured DNIS group, the DNIS group *default* is used.
2. The mapped DNIS group is checked against configured call discriminator profiles to confirm if this DNIS group/call-type combination is disallowed. If there is a match, the call is immediately rejected.
3. Once a DNIS group or a default DNIS group is identified, the customer profile associated with that DNIS group and the call type (from the bearer capability for ISDN call, statically configured for CAS calls) is selected. If there is no corresponding customer profile, the call is rejected.
4. The customer profile includes a session limit value and an overflow limit value. If these thresholds are not met, the call is then assigned the appropriate resource defined in the customer profile. If the thresholds are met, the call is rejected.

5. If resources are available from the resource group defined in the customer profile, the call is answered. Otherwise, the call is rejected.
6. As sessions start and end, the session counters increase and decrease, so the customer profile call counters are kept current.

See Figure 100 for a graphical illustration of the RPM call processes.

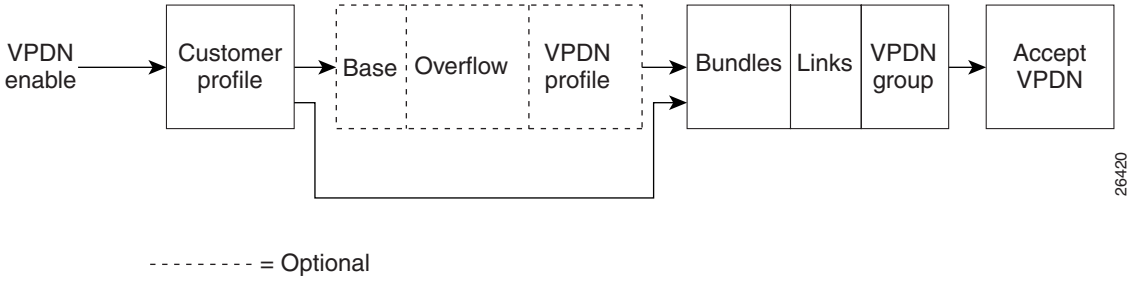
Figure 100 Incoming Call Management: RPM Functional Description



After the call is answered and if VPDN is enabled, Cisco RPM checks the customer profile for an assigned VPDN group or profile. The outgoing session management of the customer profile directs the answered call to the appropriate destination (see Figure 101), as follows:

- To a local AAA server of retail dial applications and Internet/intranet access.
- To a tunnel that is established between the NAS or LAC and a wholesale VPDN home gateway from a dial customer or LNS using L2F or L2TP tunneling technology.

Figure 101 Outgoing Call Management: RPM Functional Description for VPDN Profiles and Groups



If a VPDN profile is found, the limits are checked, as follows:

- If the limits have not been exceeded, the VPDN group data associated with that VPDN profile is used to build a VPDN tunnel.
- If the VPDN limits have been exceeded, the call is disconnected.

If a VPDN group is found within the customer profile, the VPDN group data is used to build a VPDN tunnel, as follows:

- If the VPDN group limits (number of multilink bundles, number of links per bundle) have not been exceeded, a VPDN tunnel is built.
- If the limits have been reached, the call is disconnected.

If no VPDN profile is assigned to the customer profile and VPDN is enabled, non-RPM VPDN service is attempted. If the attempt fails, the call is processed as a retail dial service call if local AAA service is available.

Accounting Data

You can generate accounting data for network dial service usage in NAS AAA attribute format.

You can configure the Cisco NAS to generate AAA accounting records for access to external AAA server option. The accounting start and stop records in AAA attribute format are sent to the external AAA server using either RADIUS server hosts or TACACS+ protocols for accounting data storage. Table 43 lists the new fields in the AAA accounting packets.

Table 43 AAA Accounting Records

Accounting Start Record	Accounting Stop Record
Call-Type	Disconnect-Cause
CAS-Group-Name	Modem-Speed-Receive
Customer-Profile-Name	Modem-Speed-Transmit
Customer-Profile-Active-Sessions	MLP-Session-ID
DNIS-Group-Name	
Overflow	
MLP-Session_ID	
Modem-Speed-Receive	
Modem-Speed-Transmit	
VPDN-Domain-Name	
VPDN-Tunnel-ID	
VPDN-HomeGateway	
VPDN-Group-Active-Sessions	

Data over Voice Bearer Services

DoVBS is a dial service that uses a customer profile and an associated resource group of digital resources to direct data calls with a speech call type to HDLC controllers.

To support ISDN DoVBS, use a DNIS group and a configured customer profile to direct the speech call to the appropriate digital resource.

The resource group assigned to this customer profile will be “digital resources” and will also have a call type of speech, so the call will terminate on an HDLC controller rather than a modem.

Call Discriminator Profiles

The Cisco RPM CLID/DNIS Call Discriminator feature lets you specify a list of calling party numbers to be rejected for inbound calls. This Cisco IOS Release 12.2 CLID/DNIS call screening feature expands previous call screening features in Cisco RPM. CLID/DNIS call screening provides an additional way to screen calls on the basis of CLID/DNIS for both local and remote RPM.

Cisco RPM CLID/DNIS Call Discriminator profiles enable you to process calls differently on the basis of the call type and CLID combination. Resource pool management offers a call discrimination feature that rejects calls on the basis of a CLID group and a call type filter. When a call arrives at the NAS, the CLID and the call type are matched against a table of disallowed calls. If the CLID and call type match entries in this table, the call is rejected before it is assigned Cisco NAS resources or before any other Cisco RPM processing occurs. This is called precall screening.

Precall screening decides whether the call is allowed to be processed. You can use the following types of discriminators to execute precall screening:

- ISDN discriminator—Accepts a call if the calling number matches a number in a group of configured numbers (ISDN group). This is also called white box screening. If you configure an ISDN group, only the calling numbers specified in the group are accepted.
- DNIS discriminator—Accepts a call if the called party number matches a number in a group of configured numbers (DNIS group). If you set up a DNIS group, only the called party numbers in the group are accepted. DNIS gives you information about the called party.
- Cisco RPM CLID/DNIS discriminator—Rejects a call if the calling number matches a number in a group of configured numbers (CLID/DNIS group). This is also called black box screening.

  If you configure a discriminator with a CLID group, the calling party numbers specified in the group are rejected. CLID gives you information about the caller.

  Similarly, if you configure a discriminator with a DNIS group, the called party numbers specified in the group are rejected.

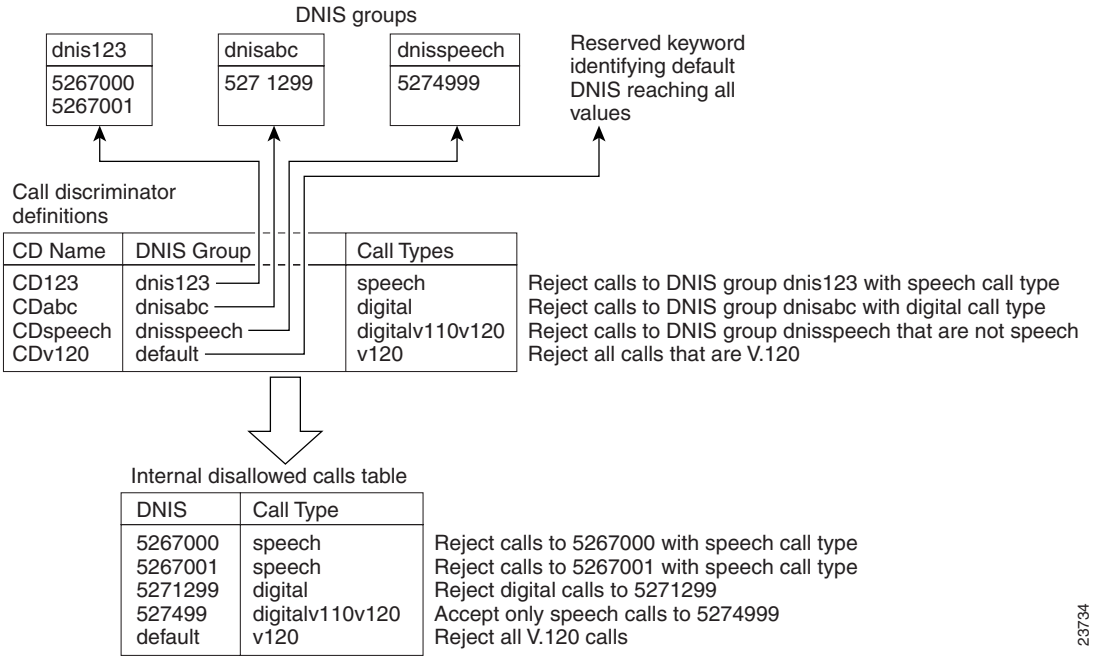
The Cisco RPM CLID/DNIS Call Discriminator Feature is independent of ISDN or DNIS screening done by other subsystems. ISDN or DNIS screening and Cisco RPM CLID/DNIS screening can both be present in the same system. Both features are executed if configured. Similarly, if DNIS Preauthorization using AAA is configured, it is present in addition to Cisco RPM CLID/DNIS screening. Refer to the *Cisco IOS Security Configuration Guide* for more information about call preauthorization.

In Cisco RPM CLID/DNIS screening, the discriminator can be a CLID discriminator, a DNIS discriminator, or a discriminator that screens on both the CLID and DNIS. The resulting discrimination logic is:

- If a discriminator contains just DNIS groups, it is a DNIS discriminator that ignores CLID. The DNIS discriminator blocks the call if the called number is in a DNIS group, which the call type references.
- If a discriminator contains just CLID groups, it is a CLID discriminator that ignores DNIS. The CLID discriminator blocks the call if the calling number is in a CLID group, which the call type references.
- If a discriminator contains both CLID and DNIS groups, it is a logical AND discriminator. It blocks the call if the calling number and called number are in the CLID or DNIS group, and the call type references the corresponding discriminator.

Figure 102 shows how call discrimination can be used to restrict a specific DNIS group to only modem calls by creating call discrimination settings for the DNIS group and the other supported call types (digital, V.110, and V.120).

Figure 102 Call Discrimination



Incoming Call Preauthentication

With ISDN PRI or channel-associated signaling (CAS), information about an incoming call is available to the NAS before the call is connected. The available call information includes:

- The DNIS, also referred to as the *called number*
- The CLID, also referred to as the *calling number*
- The call type, also referred to as the *bearer capability*

The Preauthentication with ISDN PRI and Channel-Associated Signalling feature introduced in Cisco IOS Release 12.2 allows a Cisco NAS to decide—on the basis of the DNIS number, the CLID number, or the call type—whether to connect an incoming call.

When an incoming call arrives from the public network switch, but before it is connected, this feature enables the NAS to send the DNIS number, CLID number, and call type to a RADIUS server for authorization. If the server authorizes the call, the NAS accepts the call. If the server does not authorize the call, the NAS sends a disconnect message to the public network switch to reject the call.

The Preauthentication with ISDN PRI and Channel-Associated Signalling feature offers the following benefits:

- With ISDN PRI, it enables user authentication and authorization before a call is answered. With CAS, the call must be answered; however, the call can be dropped if preauthentication fails.
- It enables service providers to better manage ports using their existing RADIUS solutions.
- Coupled with a preauthentication RADIUS server application, it enables service providers to efficiently manage the use of shared resources to offer differing service-level agreements.

For more information about the Preauthentication with ISDN PRI and Channel-Associated Signalling feature, refer to the *Cisco IOS Security Configuration Guide*.

RPM Standalone Network Access Server

A single NAS using Cisco RPM can provide the following:

- Wholesale VPDN dial service to corporate customers
- Direct remote services
- Retail dial service to end users

Figure 103 and Figure 104 show multiple connections to a Cisco AS5300 NAS. Incoming calls to the NAS can use ISDN PRI signaling, CAS, or the SS7 signaling protocol. Figure 103 shows incoming calls that are authenticated locally for retail dial services or forwarded through VPDN tunnels for wholesale dial services.

Note This implementation does not use Cisco RPM CLID/DNIS Call Discriminator Feature. If you are not using Cisco RPMS and you have more than one Cisco NAS, you must manually configure each NAS by using Cisco IOS commands. Resource usage information is not shared between NASes.

Figure 103 Retail Dial Service Using RPM

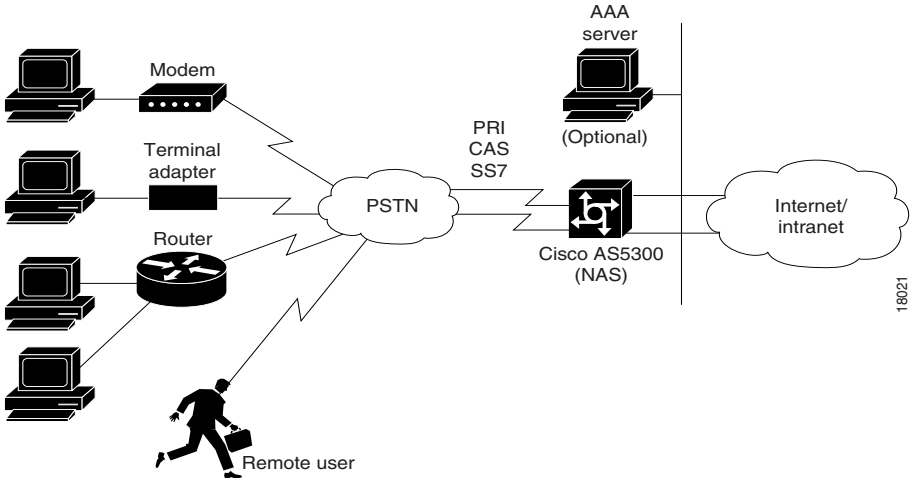
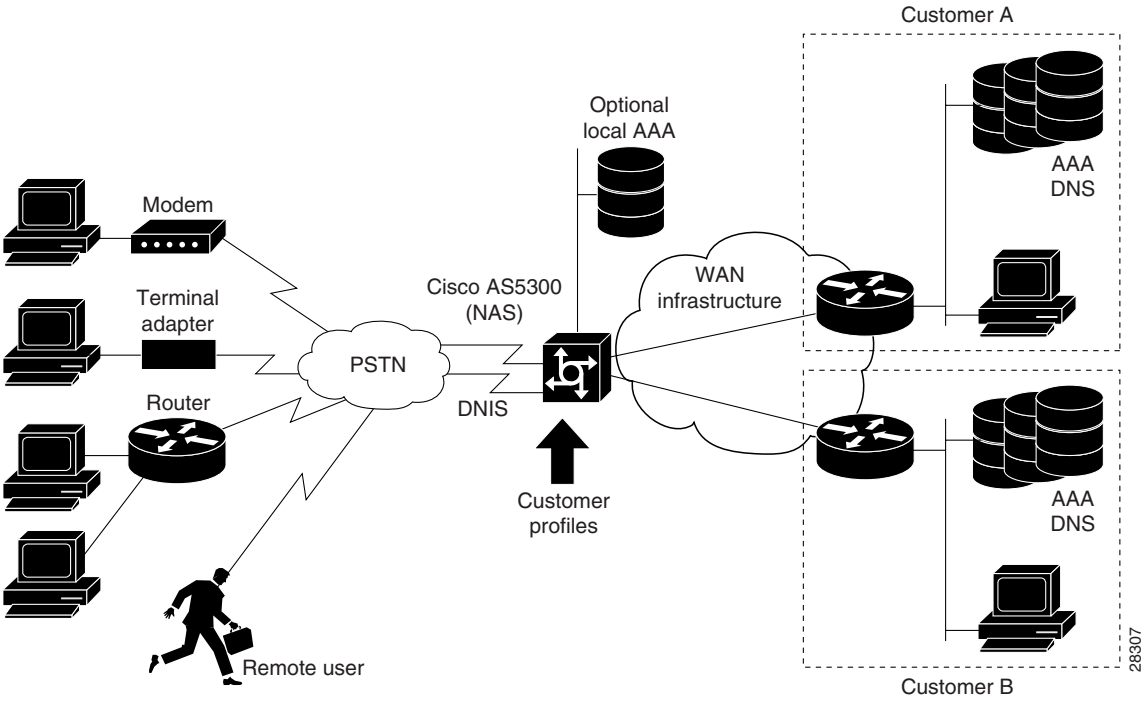


Figure 104 shows a method of implementing wholesale dial services without using VPDN tunnels by creating individual customer profiles that consist of AAA groups and PPP configurations. The AAA groups provide IP addresses of AAA servers for authentication and accounting. The PPP configurations enable you to set different PPP parameter values on each customer profile. A customer profile typically includes the following PPP parameters:

- Applicable IP address pools or a default local list of IP addresses
- Primary and secondary DNS or WINS
- Authentication method such as the Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), or Microsoft CHAP Version 1 (MS-CHAP)
- Number of links allowed for each call using Multilink PPP

Note The AAA and PPP integration applies to a single NAS environment; the external RPMS solution is not supported.

Figure 104 Resource Pool Management with Direct Remote Services



Call Processing

For call processing, incoming calls are matched to a DNIS group and the customer profile associated with that DNIS group. If a match is found, the customer profile session and overflow limits are applied and if available, the required resources are allocated. If a DNIS group is not found, the customer profile associated with the default DNIS group is used. The call is rejected if a customer profile using the default DNIS group cannot be found.

After the call is answered and if VPDN is enabled, the Cisco RPM checks the customer profile for an assigned VPDN group or profile. If a VPDN group is found, Cisco RPM authorizes VPDN by matching the group domain name or DNIS with the incoming call. If a match is found, VPDN profile session and overflow limits are applied, and, if the limits are not exceeded, tunnel negotiation begins. If the VPDN limits are exceeded, the call is disconnected.

If no VPDN profile is assigned to the customer profile and VPDN is enabled, non-RPM VPDN service will be attempted. If it fails, the call is processed as a retail dial service call if local AAA service is available.

Base Session and Overflow Session Limits

Cisco RPM enables you to set base and overflow session limits in each customer profile. The base session limit determines the maximum number of nonoverflow sessions supported for a customer profile. When the session limit is reached, if overflow sessions are not enabled, any new calls are rejected. If overflow sessions are enabled, new sessions up to the session overflow limit are processed and marked as overflow for call handling and accounting.

The session overflow limit determines the allowable number of sessions above the session limit. If the session overflow limit is greater than zero, overflow sessions are enabled and the maximum number of allowed sessions is the session limit plus the session overflow limit. While the session overflow limit has been reached, any new calls are rejected. Table 44 summarizes the effects of session and session overflow limits.

Enabling overflow sessions is useful for allocating extra sessions for preferred customers at premium rates. Overflow sessions can also be useful for encouraging customers to adequately forecast bandwidth usage or for special events when normal session usage is exceeded. For example, if a customer is having a corporate-wide program and many people are expected to request remote access, you could enable many overflow sessions and charge a premium rate for the excess bandwidth requirements.

Note An overflow call is a call received while the session limit is exceeded and is in an overflow state. When a call is identified as an overflow call, the call maintains the overflow status throughout its duration, even if the number of current sessions returns below the session limit.

Table 44 Effects of Session Limit and Session Overflow Limit Settings Combinations

Base Session Limit	Session Overflow Limit	Call Handling
0	0	Reject all calls.
10	0	Accept up to 10 sessions.
10	10	Accept up to 20 sessions and mark sessions 11 to 20 as overflow sessions.
0	10	Accept up to 10 sessions and mark sessions 1 to 10 as overflow.
All	0	Accept all calls.
0	All	Accept all calls and mark all calls as overflow.

VPDN Session and Overflow Session Limits

Cisco RPM enables you to configure base and overflow session limits per VPDN profile for managing VPDN sessions.

Note The VDPN session and session overflow limits are independent of the limits set in the customer profiles.

The base VPDN session limit determines the maximum number of nonoverflow sessions supported for a VPDN profile. When the VPDN session limit is reached, if overflow sessions are not enabled, any new VPDN calls using the VPDN profile sessions are rejected. If overflow sessions are enabled, new sessions up to the session overflow limit are processed and marked as overflow for VPDN accounting.

The VPDN session overflow limit determines the number of sessions above the session limit allowed in the VPDN group. If the session overflow limit is greater than zero, overflow sessions are enabled and the maximum number of allowed sessions is the session limit plus the session overflow limit. While the session overflow limit has been reached, any new calls are rejected.

Enabling VPDN overflow sessions is useful for allocating extra sessions for preferred customers at premium rates. Overflow sessions are also useful for encouraging customers to adequately forecast bandwidth usage or for special events when normal session usage is exceeded. For example, if a

customer is having a corporate-wide program and many people are expected to request remote access, you could enable many overflow sessions and charge a premium rate for the extra bandwidth requirements.

VPDN MLP Bundle and Links-per-Bundle Limits

To ensure that resources are not consumed by a few users with MLP connections, Cisco RPM also enables you to specify the maximum number of MLP bundles that can open in a VPDN group. In addition, you can specify the maximum number of links for each MLP bundle.

For example, if standard ISDN users access the VPDN profile, limit this setting to two links per bundle. If video conferencing is used, increase this setting to accommodate the necessary bandwidth (usually six links). These limits have no overflow option and are configured under the VPDN group component.

VPDN Tunnel Limits

For increased VPDN tunnel management, Cisco RPM enables you to set an IP endpoint session limit for each IP endpoint. IP endpoints are configured for VPDN groups.

Figure 105 and Figure 106 show logical flowcharts of RPM call processing for a standalone NAS with and without the RPM Direct Remote Services feature.

Figure 105 RPM Call-Processing Flowchart for a Standalone Network Access Server

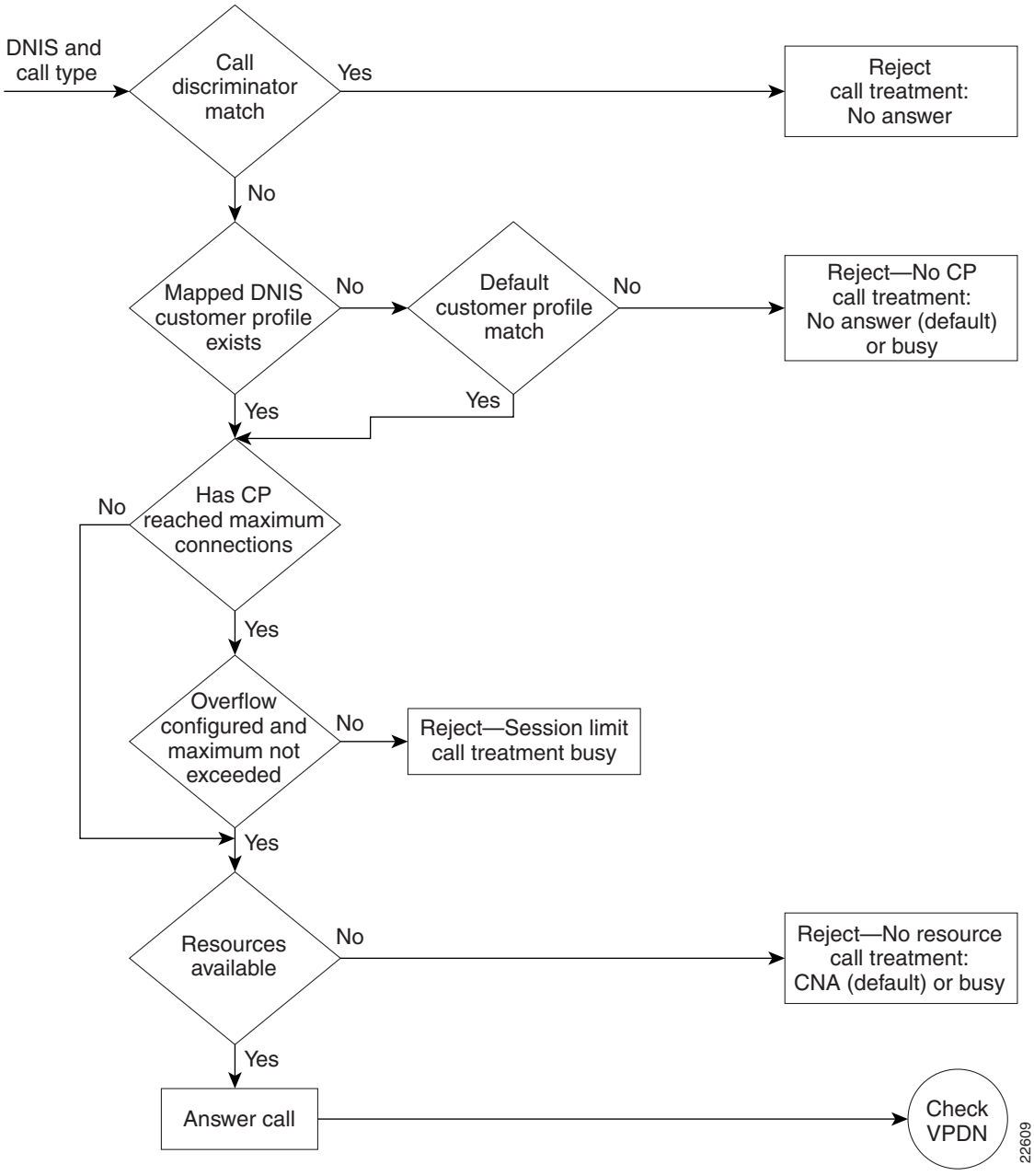
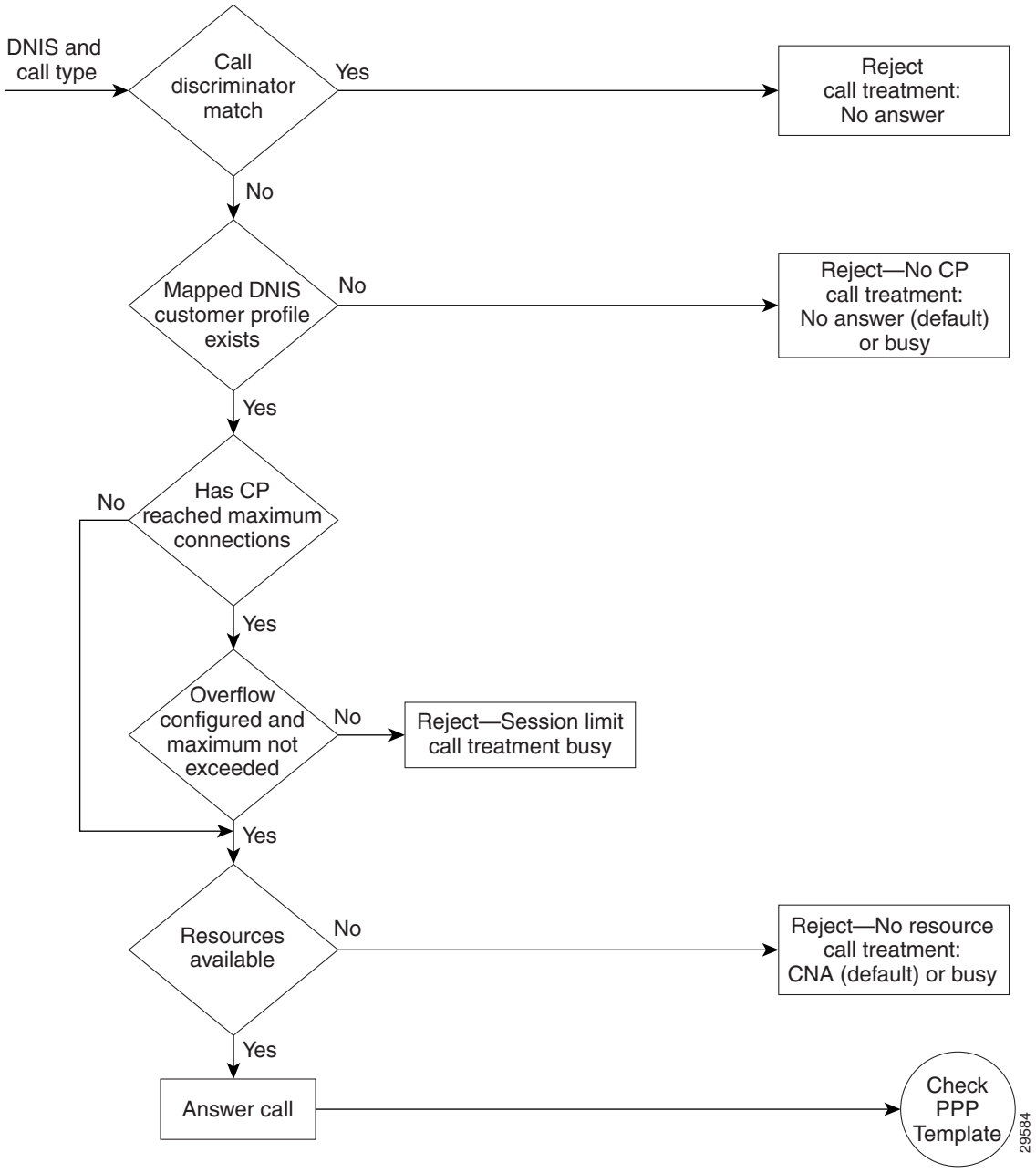


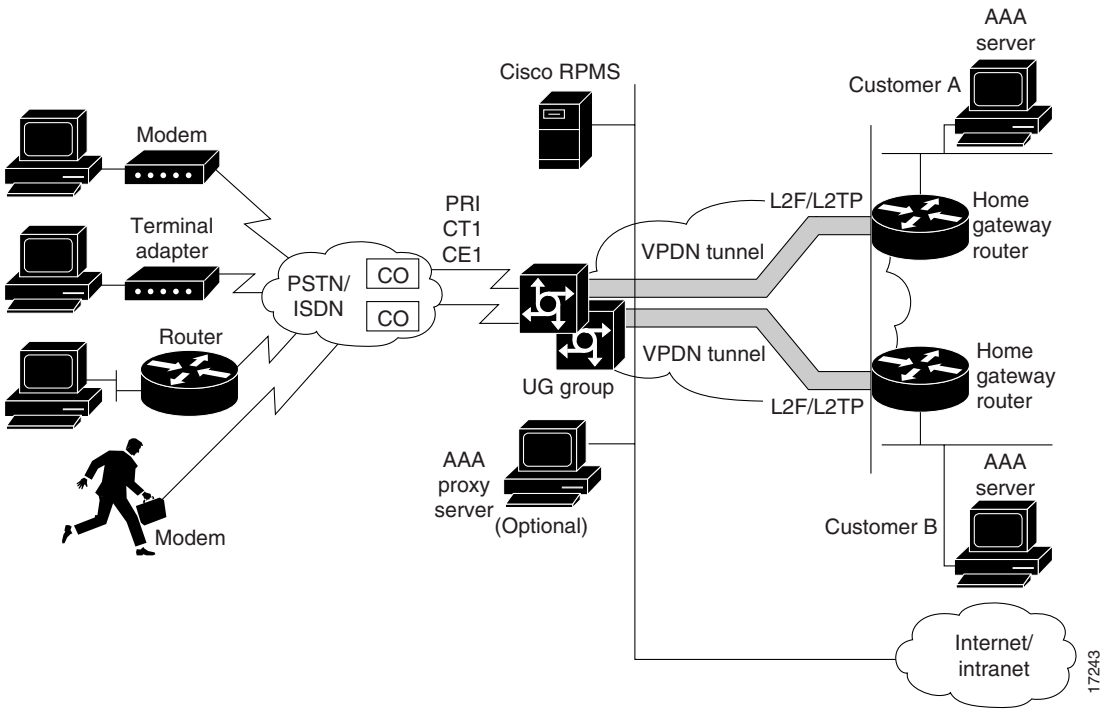
Figure 106 Flowchart for a Standalone Network Access Server with RPM Direct Remote Services



RPM Using the Cisco RPMS

Figure 107 shows a typical resource pooling network scenario using RPMS.

Figure 107 RPM Scenario Using RPMS

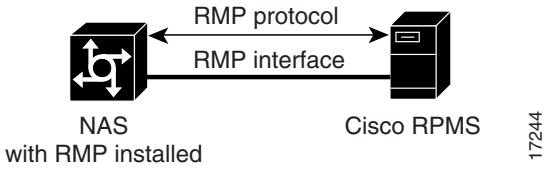


Resource Manager Protocol

Resource Manager Protocol (RMP) is a robust, recoverable protocol used for communication between the Cisco RPMS and the NAS. Each NAS client uses RMP to communicate resource management requests to the Cisco RPMS server. RPMS also periodically polls the NAS clients to query their current call information or address error conditions when they occur. RMP also allows for protocol attributes that make it extensible and enable support for customer billing requirements.

Figure 108 shows the relationship of Cisco RPM CLID/DNIS Call Discriminator Feature and RMP.

Figure 108 Cisco RPM CLID/DNIS Call Discriminator Feature and RMP



Note RMP must be enabled on all NASes that communicate with the Cisco RPM CLID/DNIS Call Discriminator Feature.

Direct Remote Services

Direct remote services is an enhancement to Cisco RPM implemented in Cisco IOS Release 12.0(7)T that enables service providers to implement wholesale dial services without using VPDN tunnels. A customer profile that has been preconfigured with a PPP template to define the unique PPP services for the wholesale dial customer is selected by the incoming DNIS and call type. At the same time, the DNIS is used to select AAA server groups for authentication/authorization and for accounting for the customer.

PPP Common Configuration Architecture (CCA) is the new component of the RPM customer profile that enables direct remote services. The full PPP command set available in Cisco IOS software is configurable per customer profile for wholesale dial applications. A customer profile typically includes the following PPP parameters:

- Local or named IP address pools
- Primary and secondary DNS or WINS addresses
- Authentication method (PAP, CHAP, MS-CHAP)
- Multilink PPP links per bundle limits

The AAA session information is selected by the incoming DNIS. AAA server lists provide the IP addresses of AAA servers for authentication, authorization, and accounting in the wholesale local network of the customer. The server lists for both authentication and authorization and for accounting contain the server addresses, AAA server type, timeout, retransmission, and keys per server.

When direct remote services is implemented on a Cisco NAS, the following sequence occurs:

1. The NAS sends an authorization request packet to the AAA server by using the authentication method (PAP, CHAP, MSCHAP) that has been configured through PPP.
2. The AAA server accepts the authorization request and returns one of the following items to the NAS:
   - A specific IP address
   - An IP address pool name
   - Nothing
3. Depending on the response from the AAA server, the NAS assigns one of the following items to the user through the DNS/WINS:
   - The IP address returned by the AAA server
   - An IP address randomly assigned from the named IP address pool
   - An IP address from a pool specified in the customer profile template

Note If the AAA server sends back to the NAS a named IP address pool and that name does not exist on the NAS, the request for service is denied. If the AAA server does not send anything back to the NAS and there is an IP address pool name configured in the customer profile template, an address from that pool is used for the session.

RPM Process with RPMS and SS7

For information on SS7 implementation for RPM, refer to the document *Cisco Resource Pool Manager Server 1.0 SS7 Implementation.*

Additional Information About Cisco RPM

For more information about Cisco RPM, see the following documents:

- *AAA Server Group*
- *Cisco Access VPN Solutions Using Tunneling Technology*
- *Cisco AS5200 Universal Access Server Software Configuration Guide*
- *Cisco AS5300 Software Configuration Guide*
- *Cisco AS5800 Access Server Software ICG*
- *Cisco Resource Pool Manager Server Configuration Guide*
- *Cisco Resource Pool Manager Server Installation Guide*
- *Cisco Resource Pool Manager Server Solutions Guide*
- *Dial Solutions Quick Configuration Guide*
- *RADIUS Multiple UDP Ports Support*
- *Redundant Link Manager*
- *Release Notes for Cisco Resource Pool Manager Server* Release 1.0
- *Resource Pool Management*
- *Resource Pool Management with Direct Remote Services*
- *Resource Pool Manager Customer Profile Template*
- *Selecting AAA Server Groups Based on DNIS*
- *SS7 Continuity Testing for Network Access Servers*
- *SS7 Dial Solution System Integration*

How to Configure RPM

Read and comply with the following restrictions and prerequisites before beginning RPM configuration:

- RPM is supported on Cisco AS5300, Cisco AS5400, and Cisco AS5800 Universal Access Servers
- Modem pooling and RPM are not compatible.
- The Cisco RPM CLID/DNIS Call Discriminator Feature must have Cisco RPM configured.
- CLID screening is not available to channel-associated signaling (CAS) interrupt level calls.
- Cisco RPM requires the NPE 300 processor when implemented on the Cisco AS5800.
- For Cisco AS5200 and Cisco AS5300 access servers, Cisco IOS Release 12.0(4)XI1 or later releases must be running on the NAS.
- For Cisco AS5800, Cisco IOS Release 12.0(5)T or later releases must be running on the NAS.
- A minimum of 64 MB must be available on the DMM cards.
- The RPM application requires an NPE 300.
- For call discriminator profiles, the Cisco AS5300, Cisco AS5400, or Cisco AS5800 Universal Access Servers require a minimum of 16 MB Flash memory and 128 MB DRAM memory, and need to be configured for VoIP as an H.323-compliant gateway.

The following tasks must be performed before configuring RPM:

- Accomplish initial configuration as described in the appropriate *Universal Access Server Software Configuration Guide*. Perform the following tasks as required.
  - Set your local AAA
  - Define your TACACS+ server for RPM
  - Define AAA accounting
  - Ensure PPP connectivity
  - Ensure VPDN connectivity

  Refer to the document *Configuring the NAS for Basic Dial Access* for more information.

To configure your NAS for RPM, perform the following tasks:

- Enabling RPM (Required)
- Configuring DNIS Groups (As required)
- Creating CLID Groups (As required)
- Configuring Discriminator Profiles (As required)
- Configuring Resource Groups (As required)
- Configuring Service Profiles (As required)
- Configuring Customer Profiles (As required)
- Configuring a Customer Profile Template (As required)
- Placing the Template in the Customer Profile (As required)
- Configuring AAA Server Groups (As required)
- Configuring VPDN Profiles (As required)
- Configuring VPDN Groups (As required)
- Counting VPDN Sessions by Using VPDN Profiles (As required)
- Limiting the Number of MLP Bundles in VPDN Groups (As required)
- Configuring Switched 56 over CT1 and RBS (As required)

See the section "Troubleshooting RPM" later in this chapter for troubleshooting tips. See the section "Configuration Examples for RPM" at the end of this chapter for examples of how to configure RPM in your network.

Enabling RPM

To enable RPM, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool enable`	Turns on RPM.
Step 2	`Router(config)# resource-pool call treatment resource channel-not-available`	Creates a resource group for resource management.
Step 3	`Router(config)# resource-pool call treatment profile no-answer`	Sets up the signal sent back to the telco switch in response to incoming calls.
Step 4	`Router(config) # resource-pool aaa protocol local`	Specifies which protocol to use for resource management.

Note If you have an RPMS, you need not define VPDN groups/profiles, customer profiles, or DNIS groups on the NAS; you need only define resource groups. Configure the remaining items by using the RPMS system.

Configuring DNIS Groups

This configuration task is optional.

To configure DNIS groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dialer dnis group` *dnis-group-name*	Creates a DNIS group. The name you specify in this step must match the name entered when configuring the customer profile.
Step 2	`Router(config-called-group)# call-type cas {digital | speech}`	Statically sets the call-type override for incoming CAS calls.
Step 3	`Router(config-called-group)# number` *number*	Enters DNIS numbers to be used in the customer profile. (Wildcards can be used.)

For default DNIS service, no DNIS group configuration is required. The following characteristics and restrictions apply to DNIS group configuration:

- Each DNIS group/call-type combination can apply to only one customer profile.
- You can use up to four default DNIS groups (one for each call type).
- You must statically configure CAS call types.
- You can use x, X or . as wildcards within each DNIS number.

Creating CLID Groups

You can add multiple CLID groups to a discriminator profile. You can organize CLID numbers for a customer or service type into a CLID group. Add all CLID numbers into one CLID group, or subdivide the CLID numbers using criteria such as call type, geographical location, or division. To create CLID groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dialer clid group clid-group-name`	Creates a CLID group, assigns it a name of up to 23 characters, and enters CLID configuration mode. The CLID group must be the same as the group specified in the customer profile configuration. Refer to the *Resource Pool Management with Direct Remote Services* document for information on configuring customer profiles.
Step 2	`Router(config-clid-group)# number clid-group-number`	Enters CLID configuration mode, and adds a CLID number to the dialer CLID group that is used in the customer profile. The CLID number can have up to 65 characters. You can use **x**, **X** or **.** as wildcards within each CLID number. The CLID screening feature rejects this number if it matches the CLID of an incoming call.

Configuring Discriminator Profiles

Discriminator profiles enable you to process calls differently on the basis of the call type and CLID/DNIS combination. The "Call Discriminator Profiles" section earlier in this chapter describes the different types of discriminator profiles that you can create.

To configure discriminator profiles for RPM implementation, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile discriminator name`	Creates a call discriminator profile and assigns it a name of up to 23 characters.
Step 2	`Router(config-call-d)# call-type {all | digital | speech | v110 | v120}`	Specifies the type of calls you want to block. The NAS will not answer the call-type you specify.

	Command	Purpose
Step 3	`Router(config-call-d)# clid group {clid-group-name | default}`	Optional. Associates a CLID group with the discriminator. If you do not specify a *clid-group-nam*e, the default discriminator in the RM is used. Any CLID number coming in on a call is in its respective default group unless it is specifically assigned a *clid-group-nam*e. After a CLID group is associated with a call type in a discriminator, it cannot be used in any other discriminator.
Step 4	`Router(config-call-d)# dnis group {dnis-group-name | default}`	Optional. Associates a DNIS group with the discriminator. If you do not specify a *dnis-group-nam*e, the default discriminator in the RM is used. Any DNIS number coming in on a call is in its respective default group unless it is specifically assigned a *dnis-group-nam*e. After a DNIS group is associated with a call type in a discriminator, it cannot be used in any other discriminator.

To verify discriminator profile settings, use the following commands:

Step 1 Use the **show resource-pool discriminator** *name* command to verify the call discriminator profiles that you configured.

If you enter the **show resource-pool discriminator** command without including a call discriminator name, a list of all current call discriminator profiles appears.

If you enter a call discriminator profile *name* with the **show resource-pool discriminator** command, the number of calls rejected by the selected call discriminator appears.

```
Router# show resource-pool discriminator

List of Call Discriminator Profiles:
    deny_CLID

Router# show resource-pool discriminator deny_CLID

    1 calls rejected
```

Step 2 Use the **show dialer** command to display general diagnostic information for interfaces configured for the dialer.

```
Router# show dialer [interface] type number
```

Configuring Resource Groups

To configure resource groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool group resource name`	Creates a resource group and assign it a name of up to 23 characters.
Step 2	`Router(config-resource-group)# range {port {slot/port slot/port}} | {limit number}`	Associates a range of modems or other physical resources with this resource group: • For port-based resources, use the physical locations of the resources. • For non-port-based resources, use a single integer limit. Specify the maximum number of simultaneous connections supported by the resource group. Up to 192 connections may be supported, depending on the hardware configuration of the access server.

For external Cisco RPMS environments, configure resource groups on the NAS before defining them on external RPMS servers.

For standalone NAS environments, first configure resource groups before using them in customer profiles.

Resource groups can apply to multiple customer profiles.

Note You can separate physical resources into groups. However, do not put heterogeneous resources in the same group. Do not put MICA technologies modems in the same group as Microcom modems. Do not put modems and HDLC controllers in the same resource group. Do not configure the **port** and **limit** command parameters in the same resource group.

Configuring Service Profiles

To configure service profiles, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile service name`	Creates a service profile and assign it a name of up to 23 characters.
Step 2	`Router(config-service-profil)# modem min-speed {speed | any} max-speed {speed | any [modulation value]}`	Specifies the desired modem parameter values. The range for **min-speed** and **max-speed** is 300 to 56000 bits per second.

Service profiles are used to configure modem service parameters for Nextport and MICA technologies modems, and support speech, digital, V.110, and V.120 call types. Error-correction and compression are hidden parameters that may be included in a service profile.

Configuring Customer Profiles

To configure customer profiles, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile customer` *name*	Creates a customer profile.
Step 2	`Router(config-customer-pro)# dnis group` {*dnis-group-name* | **default**}	Includes a group of DNIS numbers in the customer profile.
Step 3	`Router(config-customer-pro)# limit base-size` {*number* | **all**}	Specifies the base size usage limit.
Step 4	`Router(config-customer-pro)# limit overflow-size` {*number* | **all**}	Specifies the oversize size usage limit.
Step 5	`Router(config-customer-pro)# resource` *WORD* {**digital** | **speech** | **v110** | **v120**} [**service** *WORD*]	Assigns resources and supported call types to the customer profile.

Customer profiles are used so that service providers can assign different service characteristics to different customers. Note the following characteristics of customer profiles:

- Multiple resources of the same call type are used sequentially.
- The limits imposed are per customer (DNIS)—not per resource.
- A digital resource with a call type of **speech** allows for Data over Speech Bearer Service (DoSBS).

Configuring Default Customer Profiles

Default customer profiles are identical to standard customer profiles, except they do not have any associated DNIS groups. To define a default customer profile, use the reserved keyword **default** for the DNIS group:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile customer` *name*	Assigns a name to the default customer profile.
Step 2	`Router(config-customer-pro)# dnis group default`	Assigns the default DNIS group to the customer profile. This sets up the customer profile such that it will use the default DNIS configuration, which is automatically set on the NAS.

The rest of the customer profile is configured as shown in the previous section "Configuring Customer Profiles."

Configuring Customer Profiles Using Backup Customer Profiles

Backup customer profiles are customer profiles configured locally on the Cisco NAS and are used to answer calls on the basis of a configured allocation scheme when the link between the Cisco NAS and Cisco RPMS is disabled.

To enable the backup feature, you need to have already configured the following on the router:

- The **resource-pool aaa protocol group** *name* **local** command.
- All customer profiles and DNIS groups on the NAS.

The backup customer profile can contain all of the elements defined in a standard customer profile, including base size or overflow parameters. However, when the connection between the Cisco NAS and Cisco RPMS is unavailable, session counting and session limits are not applied to incoming calls. Also, after the connection is reestablished, there is no synchronization of call counters between the Cisco NAS and Cisco RPMS.

Configuring Customer Profiles for Using DoVBS

To configure customer profiles for using DoVBS, use the following commands beginning in global configuration command mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile customer` *name*	Assigns a name to a customer profile.
Step 2	`Router(config-customer-pro)# dnis group` *name*	Assigns a DNIS group to the customer profile. DNIS numbers are assigned as shown in the previous section.
Step 3	`Router(config)# limit base-size` {*number* | **all**}	Specifies the VPDN base size usage limit.
Step 4	`Router(config)# limit overflow-size` {*number* | **all**}	Specifies the VPDN overflow size usage limit.
Step 5	`Router(config-customer-pro)# resource` *name* {**digital** | **speech** | **v110** | **v120**} [**service** *name*]	Specifies resource names to use within the customer profile.

To support ISDN DoVBS, use a DNIS group and a configured customer profile to direct the speech call to the appropriate digital resource. The DNIS group assigned to the customer profile should have a call type of speech. The resource group assigned to this customer profile will be digital resources and also have a call type of speech, so the call will terminate on an HDLC controller rather than a modem.

See the section "Customer Profile Configuration for DoVBS Example" at the end of this chapter for a configuration example.

Configuring a Customer Profile Template

Customer profile templates provide a way to keep each unique situation for a customer separate for both security and accountability. This is an optional configuration task.

To configure a template and place it in a customer profile, ensure that all basic configuration tasks and the RPM configuration tasks have been completed and verified before attempting to configure the customer profile templates.

To add PPP configurations to a customer profile, create a customer profile template. Once you create the template and associate it with a customer profile by using the **source template** command, it is integrated into the customer profile.

To configure a template in RPM, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# template name`	Creates a customer profile template and assign a unique name that relates to the customer that will be receiving it. **Note** Steps 2, 3, and 4 are optional. Enter multilink, peer, and ppp commands appropriate to the application requirements of the customer.
Step 2	`Router(config-template)# peer default ip address pool pool-name`	(Optional) Specifies that the customer profile to which this template is attached will use a local IP address pool with the specified name.
Step 3	`Router(config-template)# ppp authentication chap`	(Optional) Sets the PPP link authentication method.
Step 4	`Router(config-template)# ppp multilink`	(Optional) Enables Multilink PPP for this customer profile.
Step 5	`Router(config-template)# exit`	Exits from template configuration mode; returns to global configuration mode.
Step 6	`Router(config)# resource-pool profile customer name`	Enters customer profile configuration mode for the customer to which you wish to assign this template.
Step 7	`Router(config-customer-profi)# source template name`	Attaches the customer profile template you have just configured to the customer profile.

Typical Template Configuration

The following example shows a typical template configuration:

```
template Word
  multilink {max-fragments frag-num | max-links num | min-links num}
  peer match aaa-pools
  peer default ip address {pool pool-name1 [pool-name2] | dhcp}
  ppp ipcp {dns | wins} A.B.C.D [W.X.Y.Z]
resource-pool profile customer WORD
  source template Word
  aaa group-configuration aaa-group-name

template acme_direct
  peer default ip address pool tahoe
  ppp authentication chap isdn-users
  ppp multilink
```

Verifying Template Configuration

To verify your template configuration, perform the following steps:

Step 1 Enter the **show running-config** EXEC command (where the template name is "PPP1"):

```
Router#
Router# show running-config begin template
.
.
.
```

```
template PPP1
peer default ip address pool pool1 pool2
ppp ipcp dns 10.1.1.1 10.1.1.2
ppp ipcp wins 10.1.1.3 10.1.1.4
ppp multilink max-links 2
.
.
.
```

Step 2 Ensure that your template appears in the configuration file.

Placing the Template in the Customer Profile

To place your template in the customer profile, use the following commands beginning in global configuration command mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile customer` *name*	Assigns a name to a customer profile.
Step 2	`Router(config-customer-pr)# source template`	Associates the template with the customer profile.

To verify the placement of your template in the customer profile, perform the following steps:

Step 1 Enter the **show resource-pool customer** EXEC command:

```
Router# show resource-pool customer

List of Customer Profiles:
    CP1
    CP2
```

Step 2 Look at the list of customer profiles and make sure that your profile appears in the list.

Step 3 To verify a particular customer profile configuration, enter the **show resource-pool customer** *name* EXEC command (where the customer profile name is "CP1"):

```
Router# show resource-pool customer CP1

97 active connections
    120 calls accepted
    210 max number of simultaneous connections
    50 calls rejected due to profile limits
    0 calls rejected due to resource unavailable
    90 minutes spent with max connections
    5 overflow connections
    2 overflow states entered
    0 overflow connections rejected
    0 minutes spent in overflow
    13134 minutes since last clear command
```

Configuring AAA Server Groups

To configure AAA server groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA on the NAS.

Step 2	`Router(config)# radius-server key key`
	or
	`Router(config)# tacacs-server key key`
		Set the authentication and encryption key used for all RADIUS or TACACS+ communications between the NAS and the RADIUS or TACACS+ daemon.

Step 3	`Router(config)# radius-server host {hostname | ip-address key} [auth-port port acct-port port]`
	or
	`Router(config)# tacacs-server host ip-address key`
		Specifies the host name or IP address of the server host before configuring the AAA server group. You can also specify the UDP destination ports for authentication and for accounting.

Step 4	`Router(config)# aaa group server {radius | tacacs+} group-name`	Selects the AAA server type you want to place into a server group and assign a server group name.
Step 5	`Router(config-sg radius)# server ip-address`	Specifies the IP address of the selected server type. This must be the same IP address that was assigned to the server host in Step 3.
Step 6	`Router(config-sg radius)# exit`	Returns to global configuration mode.
Step 7	`Router(config)# resource-pool profile customer name`	Enters customer profile configuration mode for the customer to which you wish to assign this AAA server group.
Step 8	`Router(config-customer-profil)# aaa group-configuration group-name`	Associates this AAA server group (named in Step 4) with the customer profile named in Step 7.

AAA server groups are lists of AAA server hosts of a particular type. The Cisco RPM currently supports RADIUS and TACACS+ server hosts. A AAA server group lists the IP addresses of the selected server hosts.

You can use a AAA server group to define a distinct list of AAA server hosts and apply this list to the Cisco RPM application. Note that the AAA server group feature works only when the server hosts in a group are of the same type.

Configuring VPDN Profiles

A VPDN profile is required only if you want to impose limits on the VPDN tunnel that are separate from the customer limits.

To configure VPDN profiles, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile vpdn profile-name`	Creates a VPDN profile and assigns it a profile name
Step 2	`Router(config-vpdn-profile)# limit base-size {number | all}`	Specifies the maximum number of simultaneous base VPDN sessions to be allowed for this VPDN group under the terms of the service-level agreement (SLA). The range is 0 to 1000 sessions. If all sessions are to be designated as base VPDN sessions, specify **all**.
Step 3	`Router(config-vpdn-profile)# limit overflow-size {number | all}`	Specifies the maximum number of simultaneous overflow VPDN sessions to be allowed for this VPDN group under the terms of the SLA. The range is 0 to 1000 sessions. If all sessions are to be designated as overflow VPDN sessions, specify **all**.
Step 4	`Router(config-vpdn-profile)# exit`	Returns to global configuration mode.
Step 5	`Router(config)# resource-pool profile customer name`	Enters customer profile configuration mode for the customer to which you wish to assign this VPDN group.

Step 6	`Router(config-customer-profi)# vpdn profile profile-name`
	or
	`Router(config-customer-profi)# vpdn group group-name`
		Attaches the VPDN profile you have just configured to the customer profile to which it belongs, or, if the limits imposed by the VPDN profile are not required, attaches VPDN group instead (see the section "Configuring VPDN Groups" later in this chapter).

Configuring VPDN Groups

To configure VPDN groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn enable`	Enables VPDN sessions on the NAS.
Step 2	`Router(config)# vpdn-group group-name`	Creates a VPDN group and assigns it a unique name. Each VPDN group can have multiple endpoints (HGW/LNSs).
Step 3	`Router(config-vpdn)# request dialin {l2f | l2tp} {ip ip-address} {domain domain-name | dnis dnis-number}`	Specifies the tunneling protocol to be used to reach the remote peer defined by a specific IP address if a dial-in request is received for the specified domain name or DNIS number. The IP address that qualifies the session is automatically generated and need not be entered again.
Step 4	`Router(config-vpdn)# multilink {bundle-number | link-number}`	Specifies the maximum number of bundles and links for all multilink users in the VPDN group. The range for both bundles and links is 0 to 32767. In general, each user requires one bundle.

	Command	Purpose
Step 5	`Router(config-vpdn)# loadsharing ip ip-address [limit number]`	Configures the endpoints for loadsharing. This router will share the load of IP traffic with the first router specified in Step 2. The **limit** keyword limits the number of simultaneous sessions that are sent to the remote endpoint (HGW/LNS). This limit can be 0 to 32767 sessions.
Step 6	`Router(config-vpdn)# backup ip ip-address [limit number] [priority number]`	Sets up a backup HGW/LNS router. The number of sessions per backup can be limited. The priority number can be 2 to 32767. The highest priority is 2, which is the first HGW/LNS router to receive backup traffic. The lowest priority, which is the default, is 32767.
Step 7	`Router(config-vpdn)# exit`	Returns to global configuration mode.

Step 8	`Router(config)# resource-pool profile vpdn profile-name`
	or
	`Router(config)# resource-pool profile customer name`
		Enters either VPDN profile configuration mode or customer profile configuration mode, depending on whether you want to allow VPDN connections for a customer profile, or allow combined session counting on all of the VPDN sessions within a VPDN profile.

Step 9	`Router(config-vpdn-profile)# vpdn group group-name`
	or
	`Router(config-customer-profi)# vpdn group group-name`
		Attaches the VPDN group to either the VPDN profile or the customer profile specified in Step 8.

A VPDN group consists of VPDN sessions that are combined and placed into a customer profile or a VPDN profile. Note the following characteristics of VPDN groups:

- The *dnis-group-name* argument is required to authorize the VPDN group with RPM.
- A VPDN group placed in a customer profile allows VPDN connections for the customer using that profile.
- A VPDN group placed in a VPDN profile allows the session limits configured for that profile to apply to all of the VPDN sessions within that VPDN group.
- VPDN data includes an associated domain name or DNIS, an endpoint IP address, the maximum number of MLP bundles, and the maximum number of links per MLP bundle; this data can optionally be located on a AAA server.

See the sections "VPDN Configuration Example" and "VPDN Load Sharing and Backing Up Between Multiple HGW/LNSs Example" at the end of this chapter for examples of using VPDN with RPM.

Counting VPDN Sessions by Using VPDN Profiles

Session counting is provided for each VPDN profile. One session is brought up each time a remote client dials into a HGW/LNS router by using the NAS/LAC. Sessions are counted by using VPDN profiles. If you do not want to count the number of VPDN sessions, do not set up any VPDN profiles. VPDN profiles count sessions in one or more VPDN groups.

To configure VPDN profile session counting, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile vpdn` *name*	Creates a VPDN profile.
Step 2	`Router(config-vpdn-profile)# vpdn-group` *name* `Router(config-vpdn-profile)# exit`	Associates a VPDN group to the VPDN profile. VPDN sessions done within this VPDN group will be counted by the VPDN profile.
Step 3	`Router(config)# resource-pool profile customer` *name* `Router(config-customer-profi)# vpdn profile` *name*	Links the VPDN group to a customer profile.
Step 4	`Router(config-customer-profi)# ^Z` `Router#`	Returns to EXEC mode to perform verification steps.

To verify session counting and view VPDN group information configured under resource pooling, use the **show resource-pool vpdn group** command. In this example, two different VPDN groups are configured under two different customer profiles:

```
Router# show resource-pool vpdn group

List of VPDN Groups under Customer Profiles
Customer Profile customer1:customer1-vpdng
Customer Profile customer2:customer2-vpdng
List of VPDN Groups under VPDN Profiles
VPDN Profile customer1-profile:customer1-vpdng
```

To display the contents of a specific VPDN group, use the **show resource-pool vpdn group** *name* command. This example contains one domain name, two DNIS called groups, and two endpoints:

```
Router# show resource-pool vpdn group customer2-vpdng

VPDN Group customer2-vpdng found under Customer Profiles: customer2

Tunnel (L2TP)
------
dnis:cg1
dnis:cg2
dnis:jan

Endpoint        Session Limit Priority Active Sessions Status Reserved Sessions
--------        ------------- -------- --------------- ------ -----------------
172.21.9.67     *             1        0               OK     -
10.1.1.1         *             2        0               OK      -
--------------- -------------           ---------------        -----------------
Total           *                      0                      0
```

To display the contents of a specific VPDN profile, use the **show resource-pool vpdn profile** *name* command, as follows:

```
Router# show resource-pool vpdn profile ?

  WORD  VPDN profile name
  <cr>

Router# show resource-pool vpdn profile customer1-profile

0 active connections
0 max number of simultaneous connections
0 calls rejected due to profile limits
```

```
0 calls rejected due to resource unavailable
0 overflow connections
0 overflow states entered
0 overflow connections rejected
1435 minutes since last clear command
```

Note Use the **debug vpdn event** command to troubleshoot VPDN profile limits, session limits, and MLP connections. First, enable this command; then, send a call into the access server. Interpret the debug output and make configuration changes as needed.

To debug the L2F or L2TP protocols, use the **debug vpdn l2x** command:

```
Router# debug vpdn l2x ?

  error            VPDN Protocol errors
  event            VPDN event
  l2tp-sequencing  L2TP sequencing
  l2x-data         L2F/L2TP data packets
  l2x-errors       L2F/L2TP protocol errors
  l2x-events       L2F/L2TP protocol events
  l2x-packets      L2F/L2TP control packets
  packet           VPDN packet
```

Limiting the Number of MLP Bundles in VPDN Groups

Cisco IOS software enables you to limit the number of MLP bundles and links supported for each VPDN group. A bundle name consists of a username endpoint discriminator (for example, an IP address or phone number) sent during LCP negotiation.

To limit the number of MLP bundles in VPDN groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn-group` *name*	Creates a VPDN group.
Step 2	`Router(config-vpdn)# multilink {bundle` *number* `| link` *number*`}`	Limits the number of MLP bundles per VPDN group and links per bundle.[1] These settings limit the number of users that can multilink.

1. Both the NAS/LAC and the HGW/LNS router must be configured to support multilink before a client can use multilink to connect to a HGW/LNS.

The following example shows the **show vpdn multilink** command output for verifying MLP bundle limits:

```
Router# show vpdn multilink

Multilink Bundle Name   VPDN Group Active links Reserved links Bundle/Link Limit
---------------------   ---------- ------------ -------------- -----------------
twv@anycompany.com      vgdnis     0            0              */*
```

Note Use the **debug vpdn event** and **debug resource-pooling** commands to troubleshoot VPDN profile limits, session limits, and MLP connections. First, enable this command; then, send a call into the access server. Interpret the debug output and make configuration changes as needed.

Configuring Switched 56 over CT1 and RBS

To configure switched 56 over CT1 and RBS, use the following commands beginning in global configuration mode. Perform this task on the Cisco AS5200 and Cisco AS5300 access servers only.

	Command	Purpose
Step 1	`Router(config)# controller t1 number`	Specifies a controller and begins controller configuration mode.
Step 2	`Router(config-controller)# cas-group 0 timeslots 1-24 type e&m-fgb {dtmf | mf} {dnis}`	Creates a CAS group and assigns time slots.
Step 3	`Router(config-controller)# framing {sf | esf}`	Specifies framing.
Step 4	`Router(config-controller)# linecode {ami | b8zs}`	Enters the line code.
Step 5	`Router(config-controller)# exit`	Returns to global configuration mode.
Step 6	`Router(config)# dialer dnis group name`	Creates a dialer called group.
Step 7	`Router(config-called-group)# call-type cas digital`	Assigns a call type as digital (switch 56).
Step 8	`Router(config-called-group)# exit`	Returns to global configuration mode.

Step 9	`Router(config)# interface serial number:number`
	
	`Router(config-if)#`
		Specifies the logical serial interface, which was dynamically created when the **cas-group** command was issued.
		
		This command also enters interface configuration mode, where you configure the core protocol characteristics for the serial interface.

To verify switched 56 over CT1, use the **show dialer dnis** command as follows:

```
Router# show dialer dnis group

List of DNIS Groups:
  default
  mdm_grp1

Router# show dialer dnis group mdm_grp1

Called Number:2001
  0 total connections
  0 peak connections
  0 calltype mismatches
Called Number:2002
  0 total connections
  0 peak connections
  0 calltype mismatches
Called Number:2003
  0 total connections
  0 peak connections
  0 calltype mismatches
Called Number:2004
  0 total connections
  0 peak connections
  0 calltype mismatches
.
.
.
```

```
Router# show dialer dnis number

List of Numbers:
  default
  2001
  2002
  2003
  2004
.
.
.
```

Verifying RPM Components

The following sections provide call-counter and call-detail output for the different RPM components:

- Verifying Current Calls
- Verifying Call Counters for a Customer Profile
- Clearing Call Counters
- Verifying Call Counters for a Discriminator Profile
- Verifying Call Counters for a Resource Group
- Verifying Call Counters for a DNIS Group
- Verifying Call Counters for a VPDN Profile
- Verifying Load Sharing and Backup

Verifying Current Calls

The following output from the **show resource-pool call** command shows the details for all current calls, including the customer profile and resource group, and the matched DNIS group:

```
Router# show resource-pool call

Shelf 0, slot 0, port 0, channel 15, state RM_RPM_RES_ALLOCATED
  Customer profile ACME, resource group isdn-ports
  DNIS number 301001
Shelf 0, slot 0, port 0, channel 14, state RM_RPM_RES_ALLOCATED
  Customer profile ACME, resource group isdn-ports
  DNIS number 301001
Shelf 0, slot 0, port 0, channel 11, state RM_RPM_RES_ALLOCATED
  Customer profile ACME, resource group MICA-modems
  DNIS number 301001
```

Verifying Call Counters for a Customer Profile

The following output from the **show resource-pool customer** command shows the call counters for a given customer profile. These counters include historical data and can be cleared.

```
Router# show resource-pool customer ACME

    3 active connections
    41 calls accepted
    3 max number of simultaneous connections
```

```
    11 calls rejected due to profile limits
    2 calls rejected due to resource unavailable
    0 minutes spent with max connections
    5 overflow connections
    1 overflow states entered
    11 overflow connections rejected
    10 minutes spent in overflow
    214 minutes since last clear command
```

Clearing Call Counters

The **clear resource-pool** command clears the call counters.

Verifying Call Counters for a Discriminator Profile

The following output from the **show resource-pool discriminator** command shows the call counters for a given discriminator profile. These counters include historical data and can be cleared.

```
Router# show resource-pool discriminator

List of Call Discriminator Profiles:
    deny_DNIS

Router# show resource-pool discriminator deny_DNIS

    1 calls rejected
```

Verifying Call Counters for a Resource Group

The following output from the **show resource-pool resource** command shows the call counters for a given resource group. These counters include historical data and can be cleared.

```
Router# show resource-pool resource

List of Resources:
    isdn-ports
    MICA-modems

Router# show resource-pool resource isdn-ports

    46 resources in the resource group
    2 resources currently active
    8 calls accepted in the resource group
    2 calls rejected due to resource unavailable
    0 calls rejected due to resource allocation errors
```

Verifying Call Counters for a DNIS Group

The following output from the **show dialer dnis** command shows the call counters for a given DNIS group. These counters include historical data and can be cleared.

```
Router# show dialer dnis group ACME_dnis_numbers

DNIS Number:301001
  11 total connections
  5 peak connections
  0 calltype mismatches
```

Verifying Call Counters for a VPDN Profile

The following output from the **show resource-pool vpdn** command shows the call counters for a given VPDN profile or the tunnel information for a given VPDN group. These counters include historical data and can be cleared.

```
Router# show resource-pool vpdn profile ACME_VPDN

        2 active connections
        2 max number of simultaneous connections
        0 calls rejected due to profile limits
        0 calls rejected due to resource unavailable
        0 overflow connections
        0 overflow states entered
        0 overflow connections rejected
        215 minutes since last clear command

Router# show resource-pool vpdn group outgoing-2

VPDN Group outgoing-2 found under VPDN Profiles:  ACME_VPDN

Tunnel (L2F)
------
dnis:301001
dnis:ACME_dnis_numbers

Endpoint    Session Limit Priority Active Sessions Status Reserved Sessions
--------    ------------- -------- --------------- ------ -----------------
172.16.1.9  *             1        2               OK     -
--------    -------------          ---------------        -----------------
Total       *                      2                      0
```

Verifying Load Sharing and Backup

The following example from the **show running-config** EXEC command shows two different VPDN customer groups:

```
Router# show running-config

Building configuration...
.
.
.
vpdn-group customer1-vpdng
 request dialin
 protocol l2f
 domain cisco.com
```

```
 domain cisco2.com
 dnis customer1-calledg
 initiate-to ip 172.21.9.67
 loadsharing ip 172.21.9.68 limit 100
 backup ip 172.21.9.69 priority 5
vpdn-group customer2-vpdng
 request dialin
 protocol l2tp
 dnis customer2-calledg
 domain acme.com
 initiate-to ip 172.22.9.5
```

Troubleshooting RPM

Test and verify that ISDN, CAS, SS7, PPP, AAA, and VPDN are working properly before implementing RPM. Once RPM is implemented, the only **debug** commands needed for troubleshooting RPM are as follows:

- **debug resource pool**
- **debug aaa authorization**

The **debug resource-pool** command is useful as a first step to ensure proper operation. It is usually sufficient for most cases. Use the **debug aaa authorization** command for troubleshooting VPDN and modem service problems.

Problems that might typically occur are as follows:

- No DNIS group found or no customer profile uses a default DNIS
- Call discriminator blocks the DNIS
- Customer profile limits exceeded
- Resource group limits exceeded

Note Always enable the debug and log time stamps when troubleshooting RPM.

This section provides the following topics for troubleshooting RPM:

- Resource-Pool Component
- Resource Group Manager
- Signaling Stack
- AAA Component
- VPDN Component
- Troubleshooting DNIS Group Problems
- Troubleshooting Call Discriminator Problems
- Troubleshooting Customer Profile Counts
- Troubleshooting Resource Group Counts
- Troubleshooting VPDN
- Troubleshooting RPMS

Resource-Pool Component

The resource-pool component contains two modules—a dispatcher and a local resource-pool manager. The dispatcher interfaces with the signaling stack, resource-group manager, and AAA, and is responsible for maintaining resource-pool call state and status information. The state transitions can be displayed by enabling the resource-pool debug traces. Table 45 summarizes the resource pooling states.

Table 45 Resource Pooling States

State	Description
RM_IDLE	No call activity.
RM_RES_AUTHOR	Call waiting for authorization; message sent to AAA.
RM_RES_ALLOCATING	Call authorized; resource group manager allocating.
RM_RES_ALLOCATED	Resource allocated; connection acknowledgment sent to signaling state. Call should get connected and become active.
RM_AUTH_REQ_IDLE	Signaling module disconnected call while in RM_RES_AUTHOR. Waiting for authorization response from AAA.
RM_RES_REQ_IDLE	Signaling module disconnected call while in RM_RES_ALLOCATING. Waiting for resource allocation response from resource group manager.

The resource-pool state can be used to isolate problems. For example, if a call fails authorization in the RM_RES_AUTHOR state, investigate further with AAA authorization debugs to determine whether the problem lies in the resource-pool manager, AAA, or dispatcher.

The resource-pool component also contains local customer profiles and discriminators, and is responsible for matching, configuring, and maintaining the associated counters and statistics. The resource-pool component is responsible for the following:

- Configuration of customer profiles or discriminators
- Matching a customer profile or discriminator for local profile configuration
- Counters/statistics for customer profiles or discriminators
- Active call information displayed by the **show resource-pool call** command

The RPMS debug commands are summarized in Table 46.

Table 46 Debug Commands for RPM

Command	Purpose
`debug resource-pool`	This debug output should be sufficient for most RPM troubleshooting situations.
`debug aaa authorization`	This debug output provides more specific information and shows the actual DNIS numbers passed and call types used.

Successful Resource Pool Connection

The following sample output from the **debug resource-pool** command displays a successful RPM connection. The entries in bold are of particular importance.

```
*Mar  1 02:14:57.439: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:21
*Mar  1 02:14:57.439: RM: event incoming call
*Mar  1 02:14:57.443: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:21
*Mar  1 02:14:57.447: RM:RPM event incoming call
*Mar  1 02:14:57.459: RPM profile ACME found
*Mar  1 02:14:57.487: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_SUCCESS
 DS0:0:0:0:21
*Mar  1 02:14:57.487: Allocated resource from res_group isdn-ports
*Mar  1 02:14:57.491: RM:RPM profile "ACME", allocated resource "isdn-ports" successfully
*Mar  1 02:14:57.495: RM state:RM_RPM_RES_ALLOCATING event:RM_RPM_RES_ALLOC_SUCCESS
DS0:0:0:0:21
*Mar  1 02:14:57.603: %LINK-3-UPDOWN: Interface Serial0:21, changed state to up
*Mar  1 02:15:00.879: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0:21, changed
state to up
```

Dialer Component

The dialer component contains DNIS groups and is responsible for configuration, and maintenance of counters and statistics. The resource-pool component is responsible for the following:

- DNIS number statistics or counters
- Configuring DNIS groups

Resource Group Manager

Resource groups are created, maintained, allocated, freed, and tallied by the resource group manager. The resource group manager is also responsible for service profiles, which are applied to resources at call setup time. The resource group manager is responsible for:

- Allocating resources when the profile has been authorized and a valid resource group is received
- Statistics or configuration of resource groups
- Configuring or applying service profiles to resource groups
- Collecting DNIS number information for channel-associated signaling calls

Signaling Stack

The signaling stacks currently supported in resource pooling are CAS and ISDN. The signaling stack delivers the incoming call to the resource-pool dispatcher and provides call-type and DNIS number information to the resource-pool dispatcher. Depending on configuration, call connect attempts may fail if the signaling stacks do not send the DNIS number and the call type to the resource-pool dispatcher. Call attempts will also fail if signaling stacks disconnect prematurely, not giving enough time for authorization or resource allocation processes to complete.

Therefore, investigate the signaling stack when call attempts or call treatment behavior does not meet expectations. For ISDN, the **debug isdn q931** command can be used to isolate errors between resource pooling, signaling stack, and switch. For CAS, the **debug modem csm**, **service internal**, and

modem-mgmt csm debug-rbs commands are used on Cisco AS5200 and Cisco AS5300 access servers, while the **debug csm** and **debug trunk cas port** *number* **timeslots** *number* commands are used on the Cisco AS5800 access server.

AAA Component

In context with resource pooling, the AAA component is responsible for the following:

- Authorization of profiles between the resource-pool dispatcher and local or external resource-pool manager
- Accounting messages between the resource-pool dispatcher and external resource-pool manager for resource allocation
- VPDN authorization between VPDN and the local or external resource-pool manager
- VPDN accounting messages between VPDN and the external resource-pool manager
- Overflow accounting records between the AAA server and resource-pool dispatcher
- Resource connect speed accounting records between the AAA server and resource group

VPDN Component

The VPDN component is responsible for the following:

- Creating VPDN groups and profiles
- Searching or matching groups based on domain or DNIS
- Maintaining counts and statistics for the groups and profiles
- Setting up the tunnel between the NAS/LAC and HGW/LNS

The VPDN component interfaces with AAA to get VPDN tunnel authorization on the local or remote resource-pool manager. VPDN and AAA debugging traces should be used for troubleshooting.

Troubleshooting DNIS Group Problems

The following output from the **debug resource-pool** command displays a customer profile that is not found for a particular DNIS group:

```
*Mar  1 00:38:21.011: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:3
*Mar  1 00:38:21.011: RM: event incoming call
*Mar  1 00:38:21.015: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:3
*Mar  1 00:38:21.019: RM:RPM event incoming call
*Mar  1 00:38:21.103: RPM no profile found for call-type digital in default DNIS number
*Mar  1 00:38:21.155: RM:RPM profile rejected do not allocate resource
*Mar  1 00:38:21.155: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_FAIL DS0:0:0:0:3
*Mar  1 00:38:21.163: RM state:RM_RPM_DISCONNECTING event:RM_RPM_DISC_ACK DS0:0:0:0:3
```

Troubleshooting Call Discriminator Problems

The following output from the **debug resource-pool** command displays an incoming call that is matched against a call discriminator profile:

```
*Mar  1 00:35:25.995: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:4
*Mar  1 00:35:25.999: RM: event incoming call
*Mar  1 00:35:25.999: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:4
*Mar  1 00:35:26.003: RM:RPM event incoming call
*Mar  1 00:35:26.135: RM:RPM profile rejected do not allocate resource
*Mar  1 00:35:26.139: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_FAIL DS0:0:0:0:4
*Mar  1 00:35:26.143: RM state:RM_RPM_DISCONNECTING event:RM_RPM_DISC_ACK DS0:0:0:0:4
```

Troubleshooting Customer Profile Counts

The following output from the **debug resource-pool** command displays what happens once the customer profile limits have been reached:

```
*Mar  1 00:43:33.275: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:9
*Mar  1 00:43:33.279: RM: event incoming call
*Mar  1 00:43:33.279: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:9
*Mar  1 00:43:33.283: RM:RPM event incoming call
*Mar  1 00:43:33.295: RPM count exceeded in profile ACME
*Mar  1 00:43:33.315: RM:RPM profile rejected do not allocate resource
*Mar  1 00:43:33.315: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_FAIL DS0:0:0:0:9
*Mar  1 00:43:33.323: RM state:RM_RPM_DISCONNECTING event:RM_RPM_DISC_ACK DS0:0:0:0:9
```

Troubleshooting Resource Group Counts

The following output from the **debug resource-pool** command displays the resources within a resource group all in use:

```
*Mar  1 00:52:34.411: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:19
*Mar  1 00:52:34.411: RM: event incoming call
*Mar  1 00:52:34.415: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:19
*Mar  1 00:52:34.419: RM:RPM event incoming call
*Mar  1 00:52:34.431: RPM profile ACME found
*Mar  1 00:52:34.455: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_SUCCESS
DS0:0:0:0:19
*Mar  1 00:52:34.459: All resources in res_group isdn-ports are in use
*Mar  1 00:52:34.463: RM state:RM_RPM_RES_ALLOCATING event:RM_RPM_RES_ALLOC_FAIL
DS0:0:0:0:19
*Mar  1 00:52:34.467: RM:RPM failed to allocate resources for "ACME"
```

Troubleshooting VPDN

Troubleshooting problems that might typically occur are as follows:

- Customer profile is not associated with a VPDN profile or VPDN group (the call will be locally terminated in this case. Regular VPDN can still succeed even if RPM/VPDN fails).
- VPDN profile limits have been reached (call answered but disconnected).
- VPDN group limits have been reached (call answered but disconnected).
- VPDN endpoint is not reachable (call answered but disconnected).

Troubleshooting RPM/VPDN Connection

The following sample output from the **debug resource-pool** command displays a successful RPM/VPDN connection. The entries in bold are of particular importance.

```
*Mar  1 00:15:53.639: Se0:10 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 00:15:53.655: RM/VPDN/ACME_VPDN: VP LIMIT/ACTIVE/RESERVED/OVERFLOW are now 6/0/0/0
*Mar  1 00:15:53.659: RM/VPDN/ACME_VPDN: Session reserved for outgoing-2
*Mar  1 00:15:53.695: Se0:10 RM/VPDN: Session has been authorized using
dnis:ACME_dnis_numbers
*Mar  1 00:15:53.695: Se0:10 RM/VPDN/session-reply: NAS name HQ-NAS
*Mar  1 00:15:53.699: Se0:10 RM/VPDN/session-reply: Endpoint addresses 172.16.1.9
*Mar  1 00:15:53.703: Se0:10 RM/VPDN/session-reply: VPDN tunnel protocol l2f
*Mar  1 00:15:53.703: Se0:10 RM/VPDN/session-reply: VPDN Group outgoing-2
*Mar  1 00:15:53.707: Se0:10 RM/VPDN/session-reply: VPDN domain dnis:ACME_dnis_numbers
*Mar  1 00:15:53.767: RM/VPDN: MLP Bundle SOHO Session Connect with 1 Endpoints:
*Mar  1 00:15:53.771:   IP 172.16.1.9 OK
*Mar  1 00:15:53.771: RM/VPDN/rm-session-connect/ACME_VPDN: VP
LIMIT/ACTIVE/RESERVED/OVERFLOW are now 6/1/0/0
*Mar  1 00:15:54.815: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0:10, changed
state to up
*Mar  1 00:15:57.399: %ISDN-6-CONNECT: Interface Serial0:10 is now connected to SOHO
```

Troubleshooting Customer/VPDN Profile

The following sample output from the **debug resource-pool** command displays when there is no VPDN group associated with an incoming DNIS group. However, the output from the **debug resource-pool** command, as shown here, does not effectively reflect the problem:

```
*Mar  1 03:40:16.483: Se0:15 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 03:40:16.515: Se0:15 RM/VPDN/rm-session-request: Authorization failed
*Mar  1 03:40:16.527: %VPDN-6-AUTHORERR: L2F NAS HQ-NAS cannot locate a AAA server for
Se0:15 user SOHO
*Mar  1 03:40:16.579: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
*Mar  1 03:40:17.539: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0:15, changed
state to up
*Mar  1 03:40:17.615: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
*Mar  1 03:40:19.483: %ISDN-6-CONNECT: Interface Serial0:15 is now connected to SOHO
```

Whenever the **debug resource-pool** command offers no further assistance besides the indication that authorization has failed, enter the **debug aaa authorization** command to further troubleshoot the problem. In this case, the **debug aaa authorization** command output appears as follows:

```
*Mar  1 04:03:49.846: Se0:19 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 04:03:49.854: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): Port='DS0:0:0:0:19'
list='default' service=RM
*Mar  1 04:03:49.858: AAA/AUTHOR/RM vpdn-session: Se0:19 (3912941997) user='301001'
*Mar  1 04:03:49.862: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
service=resource-management
*Mar  1 04:03:49.866: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
protocol=vpdn-session
*Mar  1 04:03:49.866: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
rm-protocol-version=1.0
*Mar  1 04:03:49.870: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
rm-nas-state=3278356
*Mar  1 04:03:49.874: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
rm-call-handle=27
```

```
*Mar  1 04:03:49.878: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
multilink-id=SOHO
*Mar  1 04:03:49.878: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): found list "default"
*Mar  1 04:03:49.882: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): Method=LOCAL
*Mar  1 04:03:49.886: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
service=resource-management
*Mar  1 04:03:49.890: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
protocol=vpdn-session
*Mar  1 04:03:49.890: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
rm-protocol-version=1.0
*Mar  1 04:03:49.894: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
rm-nas-state=3278356
*Mar  1 04:03:49.898: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
rm-call-handle=27
*Mar  1 04:03:49.902: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
multilink-id=SOHO
*Mar  1 04:03:49.906: Se0:19 AAA/AUTHOR/VPDN/RM/LOCAL: Customer ACME has no VPDN  group
for session dnis:ACME_dnis_numbers
*Mar  1 04:03:49.922: Se0:19 AAA/AUTHOR (3912941997): Post authorization status = FAIL
```

Troubleshooting VPDN Profile Limits

The following output from the **debug resource-pool** command displays that VPDN profile limits have been reached:

```
*Mar  1 04:57:53.762: Se0:13 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 04:57:53.774: RM/VPDN/ACME_VPDN: VP LIMIT/ACTIVE/RESERVED/OVERFLOW are now 0/0/0/0
*Mar  1 04:57:53.778: RM/VPDN/ACME_VPDN: Session outgoing-2 rejected due to Session Limit
*Mar  1 04:57:53.798: Se0:13 RM/VPDN/rm-session-request: Authorization failed
*Mar  1 04:57:53.802: %VPDN-6-AUTHORFAIL: L2F NAS HQ-NAS, AAA authorization failure for
Se0:13 user SOHO; At Session Max
*Mar  1 04:57:53.866: %ISDN-6-DISCONNECT: Interface Serial0:13  disconnected from  SOHO,
call lasted 2 seconds
*Mar  1 04:57:54.014: %LINK-3-UPDOWN: Interface Serial0:13, changed state to down
*Mar  1 04:57:54.050: RM state:RM_RPM_RES_ALLOCATED event:DIALER_DISCON DS0:0:0:0:13
*Mar  1 04:57:54.054: RM:RPM event call drop
*Mar  1 04:57:54.054: Deallocated resource from res_group isdn-ports
```

Troubleshooting VPDN Group Limits

The following **debug resource-pool** command display shows that VPDN group limits have been reached. From this display, the problem is not obvious. To troubleshoot further, use the **debug aaa authorization** command described in the "Troubleshooting RPMS" section later in this chapter:

```
*Mar  1 05:02:22.314: Se0:17 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 05:02:22.334: RM/VPDN/ACME_VPDN: VP LIMIT/ACTIVE/RESERVED/OVERFLOW are now 5/0/0/0
*Mar  1 05:02:22.334: RM/VPDN/ACME_VPDN: Session reserved for outgoing-2
*Mar  1 05:02:22.358: Se0:17 RM/VPDN/rm-session-request: Authorization failed
*Mar  1 05:02:22.362: %VPDN-6-AUTHORFAIL: L2F NAS HQ-NAS, AAA authorization failure for
Se0:17 user SOHO; At Multilink Bundle Limit
*Mar  1 05:02:22.374: %ISDN-6-DISCONNECT: Interface Serial0:17  disconnected from  SOHO,
call lasted 2 seconds
*Mar  1 05:02:22.534: %LINK-3-UPDOWN: Interface Serial0:17, changed state to down
*Mar  1 05:02:22.570: RM state:RM_RPM_RES_ALLOCATED event:DIALER_DISCON DS0:0:0:0:17
*Mar  1 05:02:22.574: RM:RPM event call drop
*Mar  1 05:02:22.574: Deallocated resource from res_group isdn-ports
```

Troubleshooting VPDN Endpoint Problems

The following output from the **debug resource-pool** command displays that the IP endpoint for the VPDN group is not reachable:

```
*Mar  1 05:12:22.330: Se0:21 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 05:12:22.346: RM/VPDN/ACME_VPDN: VP LIMIT/ACTIVE/RESERVED/OVERFLOW are now 5/0/0/0
*Mar  1 05:12:22.350: RM/VPDN/ACME_VPDN: Session reserved for outgoing-2
*Mar  1 05:12:22.382: Se0:21 RM/VPDN: Session has been authorized using
dnis:ACME_dnis_numbers
*Mar  1 05:12:22.386: Se0:21 RM/VPDN/session-reply: NAS name HQ-NAS
*Mar  1 05:12:22.386: Se0:21 RM/VPDN/session-reply: Endpoint addresses 172.16.1.99
*Mar  1 05:12:22.390: Se0:21 RM/VPDN/session-reply: VPDN tunnel protocol l2f
*Mar  1 05:12:22.390: Se0:21 RM/VPDN/session-reply: VPDN Group outgoing-2
*Mar  1 05:12:22.394: Se0:21 RM/VPDN/session-reply: VPDN domain dnis:ACME_dnis_numbers
*Mar  1 05:12:25.762: %ISDN-6-CONNECT: Interface Serial0:21 is now connected to SOHO
*Mar  1 05:12:27.562: %VPDN-5-UNREACH: L2F HGW 172.16.1.99 is unreachable
*Mar  1 05:12:27.578: RM/VPDN: MLP Bundle SOHO Session Connect with 1 Endpoints:
*Mar  1 05:12:27.582:   IP 172.16.1.99 Destination unreachable
```

Troubleshooting RPMS

In general, the **debug aaa authorization** command is not used for RPM troubleshooting unless the **debug resource-pool** command display is too vague. The **debug aaa authorization** command is more useful for troubleshooting with RPMS. Following is sample output:

```
Router# debug aaa authorization

AAA Authorization debugging is on

Router# show debug

General OS:
  AAA Authorization debugging is on
Resource Pool:
  resource-pool general debugging is on
```

The following output from the **debug resource-pool** and **debug aaa authorization** commands shows a successful RPM connection:

```
*Mar 1 06:10:35.450: AAA/MEMORY: create_user (0x723D24) user='301001'
ruser=''port='DS0:0:0:0:12' rem_addr='102' authen_type=NONE service=NONE priv=0
*Mar 1 06:10:35.462: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907):
Port='DS0:0:0:0:12' list='default' service=RM
*Mar 1 06:10:35.466: AAA/AUTHOR/RM call-accept: DS0:0:0:0:12 (2784758907) user= '301001'
*Mar 1 06:10:35.470: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
service=resource-management
*Mar 1 06:10:35.470: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
protocol=call-accept
*Mar 1 06:10:35.474: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-protocol-version=1.0
*Mar 1 06:10:35.478: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-nas-state=7513368
*Mar 1 06:10:35.482: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-call-type=speech
*Mar 1 06:10:35.486: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-request-type=dial-in
*Mar 1 06:10:35.486: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-link-type=isdn
```

```
*Mar  1 06:10:35.490: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): found list
"default"
*Mar 1 06:10:35.494: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): Method=LOCAL
*Mar 1 06:10:35.498: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907):Received DNIS=301001
*Mar 1 06:10:35.498: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907):Received CLID=102
*Mar 1 06:10:35.502: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907):Received
Port=DS0:0:0:0:12
*Mar 1 06:10:35.506: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
service=resource-management
*Mar 1 06:10:35.510: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
protocol=call-accept
*Mar 1 06:10:35.510: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-protocol-version=1.0
*Mar 1 06:10:35.514: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-nas-state=7513368
*Mar 1 06:10:35.518: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-call-type=speech
*Mar 1 06:10:35.522: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-request-type=dial-in
*Mar 1 06:10:35.526: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-link-type=isdn
*Mar 1 06:10:35.542: AAA/AUTHOR (2784758907): Post authorization status = PASS_REPL
*Mar 1 06:10:35.546: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
service=resource-management
*Mar 1 06:10:35.550: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
protocol=call-accept
*Mar 1 06:10:35.554: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-protocol-version=1.0
*Mar 1 06:10:35.558: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-response-code=overflow
*Mar 1 06:10:35.558: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-call-handle=47
*Mar 1 06:10:35.562: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-call-count=2
*Mar 1 06:10:35.566: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-cp-name=ACME
*Mar 1 06:10:35.570: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-rg-name#0=MICA-modems
*Mar 1 06:10:35.574: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-rg-service-name#0=gold
*Mar 1 06:10:35.578: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-call-treatment=busy
*Mar  1 06:10:35.582: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-call-type=speech
```

Configuration Examples for RPM

The following sections provide RPM configuration examples:

- Standard Configuration for RPM Example
- Customer Profile Configuration for DoVBS Example
- DNIS Discriminator Profile Example
- CLID Discriminator Profile Example
- Direct Remote Services Configuration Example
- VPDN Configuration Example
- VPDN Load Sharing and Backing Up Between Multiple HGW/LNSs Example

Standard Configuration for RPM Example

The following example demonstrates a basic RPM configuration:

```
resource-pool enable
resource-pool call treatment resource busy
resource-pool call treatment profile no-answer
!
resource-pool group resource isdn-ports
 range limit 46
resource-pool group resource MICA-modems
 range port 1/0 2/23
!
resource-pool profile customer ACME
 limit base-size 30
 limit overflow-size 10
 resource isdn-ports digital
 resource MICA-modems speech service gold
 dnis group ACME_dnis_numbers
!
resource-pool profile customer DEFAULT
 limit base-size 10
 resource MICA-modems speech service silver
 dnis group default

resource-pool profile discriminator deny_DNIS
 call-type digital
 dnis group bye-bye
!
resource-pool profile service gold
 modem min-speed 33200 max-speed 56000 modulation v90
resource-pool profile service silver
 modem min-speed 19200 max-speed 33200 modulation v34
!
resource-pool aaa protocol local
!
dialer dnis group ACME_dnis_numbers
 number 301001
dialer dnis group bye-bye
 number 301005
```

Tips

- Replace the command string **resource isdn-ports digital** in the previous example with **resource isdn-ports speech** to set up DoVBS. See the section, "Customer Profile Configuration for DoVBS Example," for more information.

  Digital calls to 301001 are associated with the customer ACME by using the resource group "isdn-ports."
- Speech calls to 301001 are associated with the customer ACME by using the resource group "mica-modems" and allow for V.90 connections (anything less than V.90 is also allowed).
- Digital calls to 301005 are denied.
- All other speech calls to any other DNIS number are associated with the customer profile "DEFAULT" by using the resource group "mica-modems" and allow for V.34 connections (anything more than V.34 is not allowed; anything less than V.34 is also allowed).
- All other digital calls to any other DNIS number are not associated with a customer profile and are therefore not allowed.

- The customer profile named "DEFAULT" serves as the default customer profile for speech calls only. If the solution uses an external RPMS server, this same configuration can be used for backup resource pooling if communication is lost between the NAS and the RPMS.

Customer Profile Configuration for DoVBS Example

To allow ISDN calls with a speech bearer capability to be directed to digital resources, make the following change (highlighted in bold) to the configuration shown in the previous section, "Standard Configuration for RPM Example":

```
resource-pool profile customer ACME
 limit base-size 30
 limit overflow-size 10
 resource isdn-ports speech
 dnis group ACME_dnis_numbers
```

This change causes ISDN speech calls (in addition to ISDN digital calls) to be directed to the resource "isdn-ports"; thus, ISDN speech calls provide DoVBS.

DNIS Discriminator Profile Example

The following is sample configuration for a DNIS discriminator. It shows how to enable resource pool management, configure a customer profile, create DNIS groups, and add numbers to the DNIS groups.

```
aaa new-model
!
! Enable resource pool management
resource-pool enable
!
resource-pool group resource digital
 range limit 20
!
! Configure customer profile
resource-pool profile customer cp1
 limit base-size all
 limit overflow-size 0
 resource digital digital
 dnis group ok
!
!
isdn switch-type primary-5ess
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback1
 ip address 192.168.0.0 255.255.255.0
!
interface Serial0:23
 ip unnumbered Loopback1
 encapsulation ppp
 ip mroute-cache
 dialer-group 1
 isdn switch-type primary-5ess
```

```
 no peer default ip address
 ppp authentication chap
!
! Configure DNIS groups
dialer dnis group blot
 number 5552003
 number 3456789
 number 2345678
 number 1234567
!
dialer dnis group ok
 number 89898989
 number 5551003
!
dialer-list 1 protocol ip permit
```

CLID Discriminator Profile Example

The following is a sample configuration of a CLID discriminator. It shows how to enable resource pool management, configure resource groups, configure customer profiles, configure CLID groups and DNIS groups, and add them to discriminator profiles.

```
version xx.x
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname cisco-machine
!
aaa new-model
aaa authentication login djm local
!
username eagle password ***
username infiniti password ***
spe 1/0 1/7
 firmware location system:/ucode/mica_port_firmware
spe 2/0 2/7
 firmware location system:/ucode/mica_port_firmware
!
! Enable resource pool management
resource-pool enable
!
! Configure resource groups
resource-pool group resource digital
 range limit 20
!
! Configure customer profiles
resource-pool profile customer cp1
 limit base-size all
 limit overflow-size 0
 resource digital digital
dnis group ok
!
! Configure discriminator profiles
resource-pool profile discriminator baadaabing
 call-type digital
clid group stompIt
!
```

```
resource-pool profile discriminator baadaaboom
 call-type digital
 clid group splat
!
ip subnet-zero
!
isdn switch-type primary-5ess
chat-script dial ABORT BUSY "" AT OK "ATDT \T" TIMEOUT 30 CONNECT \c
!
!
mta receive maximum-recipients 0
partition flash 2 8 8
!
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 1
 shutdown
 clock source line secondary 1
!
controller T1 2
 shutdown
 clock source line secondary 2
!
controller T1 3
 shutdown
 clock source line secondary 3
!
controller T1 4
 shutdown
 clock source line secondary 4
!
controller T1 5
 shutdown
 clock source line secondary 5
!
controller T1 6
 shutdown
 clock source line secondary 6
!
controller T1 7
 shutdown
 clock source line secondary 7
!
interface Loopback0
 ip address 192.168.12.1 255.255.255.0
!
interface Loopback1
 ip address 192.168.15.1 255.255.255.0
!
interface Loopback2
 ip address 192.168.17.1 255.255.255.0
!
interface Ethernet0
 ip address 10.0.39.15 255.255.255.0
 no ip route-cache
 no ip mroute-cache
!
```

```
interface Serial0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no fair-queue
 clockrate 2015232
!
interface Serial1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no fair-queue
 clockrate 2015232
!
interface Serial2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no fair-queue
 clockrate 2015232
!
interface Serial3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no fair-queue
 clockrate 2015232
!
interface Serial0:23
 ip unnumbered Loopback1
 encapsulation ppp
 ip mroute-cache
 dialer-group 1
 isdn switch-type primary-5ess
 no peer default ip address
 ppp authentication chap pap
!
interface FastEthernet0
 ip address 10.0.38.15 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex half
 speed 100
!
!
ip local pool default 192.168.13.181 192.168.13.226
ip classless
ip route 172.25.0.0 255.0.0.0 Ethernet0
ip route 172.19.0.0 255.0.0.0 Ethernet0
no ip http server
!
!
! Configure DNIS groups
dialer dnis group blot
 number 4085551003
 number 5552003
 number 2223333
 number 3456789
 number 2345678
 number 1234567
```

```
!
dialer dnis group ok
 number 89898989
 number 4084442002
 number 4085552002
 number 5551003
!
dialer clid group splat
 number 12321224
!
! Configure CLID groups
dialer clid group zot
 number 2121212121
 number 4085552002
!
dialer clid group snip
 number 1212121212
!
dialer clid group stompIt
 number 4089871234
!
dialer clid group squash
 number 5656456
dialer-list 1 protocol ip permit
!
!
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
line 1 96
 no exec
 exec-timeout 0 0
 autoselect ppp
line aux 0
line vty 0 4
 exec-timeout 0 0
transport input none
!
scheduler interval 1000
end
```

Direct Remote Services Configuration Example

The following example shows a direct remote services configuration:

```
resource-pool profile customer ACME
 limit base-size 30
 limit overflow-size 10
 resource isdn-ports digital
 resource MICA-modems speech service gold
 dnis group ACME_dnis_numbers
 aaa group-configuration tahoe
 source  template acme_direct
!
resource-pool profile customer DEFAULT
 limit base-size 10
 resource MICA-modems speech service silver
 dnis group default
```

```
resource-pool profile discriminator deny_DNIS
  call-type digital
  dnis group bye-bye
!
resource-pool profile service gold
 modem min-speed 33200 max-speed 56000 modulation v90
resource-pool profile service silver
 modem min-speed 19200 max-speed 33200 modulation v34
!
resource-pool aaa protocol local
!
template acme_direct
 peer default ip address pool tahoe
 ppp authentication chap isdn-users
 ppp multilink
!
dialer dnis group ACME_dnis_numbers
 number 301001
dialer dnis group bye-bye
 number 301005
```

VPDN Configuration Example

Adding the following commands to those listed in the section "Standard Configuration for RPM Example" earlier in this chapter allows you to use VPDN by setting up a VPDN profile and a VPDN group:

Note If the limits imposed by the VPDN profile are not required, do not configure the VPDN profile. Replace the **vpdn profile** *ACME_VPDN* command under the customer profile ACME with the **vpdn group** *outgoing-2* command.

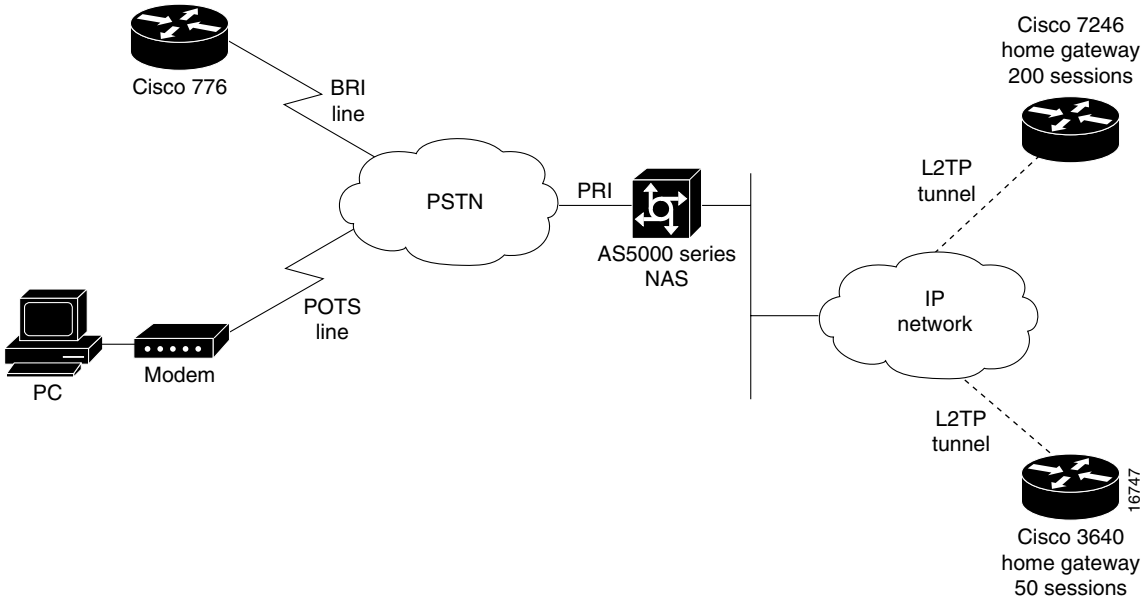
```
resource-pool profile vpdn ACME_VPDN
 limit base-size 6
 limit overflow-size 0
 vpdn group outgoing-2
!
resource-pool profile customer ACME
 limit base-size 30
 limit overflow-size 10
 resource isdn-ports digital
 resource MICA-modems speech service gold
 dnis group ACME_dnis_numbers
!
vpdn profile ACME_VPDN
!
vpdn enable
!
vpdn-group outgoing-2
 request dialin
  protocol l2f
  dnis ACME_dnis_numbers
 local name HQ-NAS
 initiate-to ip 172.16.1.9
 multilink bundle 1
 multilink link 2
!
dialer dnis group ACME_dnis_numbers
 number 301001
```

VPDN Load Sharing and Backing Up Between Multiple HGW/LNSs Example

Cisco IOS software enables you to balance and back up VPDN sessions across multiple tunnel endpoints (HGW/LNS). When a user or session comes into the NAS/LAC, a VPDN load-balancing algorithm is triggered and applied to the call. The call is then passed to an available HGW/LNS. You can modify this function by limiting the number of sessions supported on an HGW/LNS router and limiting the number of MLP bundles and links.

Figure 109 shows an example of one NAS/LAC that directs calls to two HGW/LNS routers by using the L2TP tunneling protocol. Each router has a different number of supported sessions and works at a different speed. The NAS/LAC is counting the number of active simultaneous sessions sent to each HGW/LNS.

Figure 109 Home Gateway Load Sharing and Backup



In a standalone NAS environment (no RPMS server used), the NAS has complete knowledge of the status of tunnel endpoints. Balancing across endpoints is done by a “least-filled tunnel” or a “next-available round robin” approach. In an RPMS-controlled environment, RPMS has the complete knowledge of tunnel endpoints. However, the NAS still has the control over those tunnel endpoints selected by RPMS.

A standalone NAS uses the following default search criteria for load-balancing traffic across multiple endpoints (HGW/LNS):

- Select any idle endpoint—an HGW/LNS with no active sessions.
- Select an active endpoint that currently has a tunnel established with the NAS.
- If all specified load-sharing routers are busy, select the backup HGW. If all endpoints are busy, report that the NAS cannot find an IP address to establish the call.

Note This default search order criteria is independent of the Cisco RPMS application scenario. A standalone NAS uses a different load-sharing algorithm than the Cisco RPMS. This search criteria will change as future enhancements become available.

The following is an example of VPDN load sharing between multiple HGW/LNSs:

```
vpdn enable
!
vpdn-group outgoing-2
 request dialin
  protocol l2tp
  dnis ACME_dnis_numbers
 local name HQ-NAS
 initiate-to ip 172.16.1.9
 loadsharing ip 172.16.1.9 limit 200
 loadsharing ip 172.16.2.17 limit 50
 backup ip 172.16.3.22
```

Configuring Wholesale Dial Performance Optimization

This chapter describes the Wholesale Dial Performance Optimization feature in the following sections:

- Wholesale Dial Performance Optimization Feature Overview
- How to Configure Automatic Command Execution
- How to Configure TCP Clear Performance Optimization
- Verifying Configuration of TCP Clear Performance Optimization

Note This task provides inbound and outbound performance optimization for wholesale dial customers who provide ports to America Online (AOL). It is configured only on Cisco AS5800 access servers.

Wholesale Dial Performance Optimization Feature Overview

Both the inbound and outbound aspects of this feature are enabled using the **autocommand-options telnet-faststream** command.

- Outbound—Provides stream processing, allowing the output data processing to occur at the interrupt level. Being event driven, this removes polling and process switching overhead. In addition, the flow control algorithm is enhanced to handle the higher volume of traffic and to eliminate some out-of-resource conditions that could result in abnormal termination of the session.
- Inbound—Provides stream processing with the same improvements as for outbound traffic. Also, it removes scanning for special escape characters in the data stream; this is very process-intensive and is not required for this application. (In other situations, the escape characters allow for a return to the privileged EXEC mode prompt (#) on the router.) In addition, Nagle's algorithm is used to form the inbound data stream into larger packets, thus minimizing packet-processing overhead.

This configuration is designed to provide more efficiency in the data transfers for AOL port suppliers who are using a Cisco network access server to communicate with a wholesale dial carrier.

The Cisco AS5800 access server is required to support all dial-in lines supported by two complete T3 connections (that is, 1344 connections) running TCP Clear connections to an internal host. The desired average data throughput for these connections is 6 kbps outbound and 3 kbps inbound.

When using the **autocommand-options telnet-faststream** command, no special character processing, including break recognition, is performed on incoming data from the dial shelf. This requires the TCP Clear connection to run as the sole connection on the TTY line. This sole connection is terminated by TTY line termination or TCP connection termination, with no EXEC session capability for the user. This

has been implemented by specifying a new **autocommand-options telnet-faststream** command that, in conjunction with the **autocommand telnet** command with the **/stream** option, enables Telnet faststream processing. This capability is also available for TACACS/RADIUS attribute-value pair processing, because this processing uses the **autocommand** facility.

How to Configure Automatic Command Execution

The following are three options for configuring the **autocommand telnet /stream** line configuration command:

- Automatic command execution can be configured on the lines.
- Automatic command execution can be configured using user ID and password.
- Automatic command execution can also be configured at a TACACS/RADIUS server, if the username authentication is to be performed there, rather than on the router.

To configure automatic command execution on the lines of a Cisco AS5800 universal network access server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line 1/3/00 1/11/143`	Selects the lines to be configured and begins line configuration mode.
Step 2	`Router(config-line)# autocommand telnet aol-host 5190 /stream`	Configures autocommand on the lines.

To configure automatic command execution using a user ID and password on a Cisco AS5800 universal network access server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# username aol password aol`	Defines the user ID and password.
Step 2	`Router(config)# username aol autocommand telnet aol-host 5190 /stream`	Configures autocommand on the user ID.

You can also configure automatic command execution at a TACACS/RADIUS server if the username authentication is to be performed there rather than on the router. The AV-pair processing allows autocommand to be configured.

How to Configure TCP Clear Performance Optimization

To enable TCP Clear performance optimization, automatic command execution must be configured to enable Telnet faststream capability. To implement TCP Clear performance optimization on a Cisco AS5800 universal network access server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line 1/3/00 1/11/143`	Selects the lines to be configured and begins line configuration mode.
Step 2	`Router(config-line)# autocommand telnet-faststream`	Enables the TCP Clear performance optimization on the selected lines.

Verifying Configuration of TCP Clear Performance Optimization

To check for correct configuration, use the **show line** command. In the following example, Telnet faststream is enabled under "Capabilities".

```
Router# show line 1/4/00

   Tty Typ     Tx/Rx    A Modem  Roty AccO AccI   Uses   Noise  Overruns   Int
*   1/4/00 Digital modem - inout     -    -    -      1       0     0/0       -

Line 1/4/00, Location: "", Type: ""
Length: 24 lines, Width: 80 columns
Status: PSI Enabled, Ready, Connected, Active, No Exit Banner
  Modem Detected
Capabilities: Hardware Flowcontrol In, Hardware Flowcontrol Out
  Modem Callout, Modem RI is CD, Line usable as async interface
  Hangup on Last Close, Modem Autoconfigure, Telnet Faststream
Modem state: Ready
Modem hardware state: CTS DSR  DTR RTS
modem=1/4/00, vdev_state(0x00000000)=CSM_OC_STATE, bchan_num=(T1 1/2/0:7:20)
vdev_status(0x00000001): VDEV_STATUS_ACTIVE_CALL.

Group codes:    0, Modem Configured
Special Chars: Escape  Hold  Stop  Start  Disconnect  Activation
                ^^x    none   -     -       none
Timeouts:      Idle EXEC    Idle Session   Modem Answer  Session   Dispatch
                never         never                      none     not set
                            Idle Session Disconnect Warning
                              never
                            Login-sequence User Response
                             00:00:30
                            Autoselect Initial Wait
                              not set
Modem type is 9600.
Session limit is not set.
Time since activation: never
Editing is enabled.
History is enabled, history size is 10.
DNS resolution in show commands is enabled
Full user help is disabled
Allowed transports are telnet.  Preferred is lat.
Automatically execute command "telnet 10.100.254.254 2145 /stream"
No output characters are padded
```

Modem Initialization Strings

This appendix provides tables that contain modem initialization strings and sample modem initialization scripts. Table 50 lists required settings, and error compression (EC) and compression settings for specific modem types. Use this information to create your modem scripts. Table 51 lists information for setting AUX ports. SeeTable 52 for a legend of symbols used in these two tables. Sample scripts follow the tables.

For information about configuring lines to support modems, see the chapters in the part "Modem and Dial Shelf Configuration and Management" in this publication.

Table 50 Required Settings and EC/Compression Settings

Settings Required for All Modems					**Settings for EC/Compression**					
Modem	**FD**	**AA**	**CD**	**DTR**	**RTS/CTS Flow**	**LOCK DTE Speed**	**Best Error**	**Best Comp**	**No Error**	**No Comp**
Codex 3260	&F	S0=1	&C1	&D3	*FL3	*SC1	*SM3	*DC1	*SM1	*DC0
USR Courier USR Sportster	&F	S0=1	&C1	&D3	&H1&R 2	&B1	&M4	&K1	&M0	&K0
Global Village Teleport Gold	&F	S0=1	&C1	&D3	\Q3	\J0	\N7	%C1	\N0	%C0
Telebit T1600/T3000/ WB	&F1	S0=1	&C1	&D3	S58=2 S68=2	S51=6	S180=2 S181=1	S190=1	S180=0 S181=1	S190=0
Telebit T2500 (ECM)	&F	S0=1	&C1	&D3	S58=2 S68=2	S51=6	S95=2	S98=1 S96=1	S95=0	S98=0 S96=0
Telebit Trailblazer	&F	S0=1	&C1							
AT&T Paradyne Dataport	&F	S0=1	&C1	&D3	\Q3	--->	\N7	%C1	\N0	%C0
Hayes modems Accura/ Optima	&F	S0=1	&C1	&D3	&K3	&Q6	&Q5	&Q9	&Q6	<---
Microcom QX4232 series	&F	S0=1	&C1	&D3	\Q3	\J0	\N6	%C1	\N0	%C0

Table 50 Required Settings and EC/Compression Settings (continued)

Settings Required for All Modems					Settings for EC/Compression					
Modem	**FD**	**AA**	**CD**	**DTR**	**RTS/CTS Flow**	**LOCK DTE Speed**	**Best Error**	**Best Comp**	**No Error**	**No Comp**
Motorola UDS FastTalk II	&F	S0=1	&C1	&D3	\Q3	\J0	\N6	%C1	\N0	%C0
Multitech MT1432 MT932	&F	S0=1	&C1	&D3	&E4	$BA0	&E1	&E15	&E0	&E14
Digicom Scout Plus	&F	S0=1	&C1	&D3	*F3	*S1	*E9	<---	*E0	<---
Digicom SoftModem	&F	S0=1	&C1	&D3	&K3	--->	\N5	%C1	\N0	%C0
Viva 14.4/9642c	&F	S0=1	&C1	&D3	&K3	--->	\N3	%M3	\N0	%M0
ZyXel U-1496E	&F	S0=1	&C1	&D3	&H3	&B1	&K4	<---	&K0	<---
Supra V.32bis/28.8	&F	S0=1	&C1	&D3	&K3	--->	\N3	%C1	\N0	%C0
ZOOM 14.4	&F	S0=1	&C1	&D3	&K3	--->	\N3	%C2	\N0	%C0
Intel External	&F	S0=1	&C1	&D3	\Q3	\J0	\N3	%C1"H 3	\N0	%C0
Practical Peripherals	&F	S0=1	&C1	&D3	&K3	--->	&Q5	&Q9	&Q6	<---

Table 51 AUX and Platform Specific Settings

Modem	Settings for Use with AUX Port		Other Settings		Comments
	No Echo	No Res	CAB-MDCE	Write Memory	
Codex 3260	E0	Q1	&S1	&W	
USR Courieræ USR Sportster	E0	Q1	*NA*	&W	
Global Village Teleport Gold	E0	Q1	*NA*	&W	
Telebit T1600/T3000/ WB	E0	Q1	&S4	&W	All Telebit modems need to have the speed set explicitly. These examples use 38400 bps. Using what Telebit calls "UNATTENDED ANSWER MODE" is the best place to start a dial in only modem.
Telebit T2500 (ECM)	E0	Q1	&S1	&W	
Telebit Trailblazer	E0	Q1	*NA*	&W	Use "ENHANCED COMMAND MODE" on the T2500.
AT&T Paradyne Dataport	E0	Q1	*NA*	&W	Almost all Microcom modems have similar configuration parameters.
Hayes modems Accura/ Optima	E0	Q1	*NA*	&W	
Microcom QX4232 series	E0	Q1	*NA*	&W	
Motorola UDS FastTalk II	E0	Q1	*NA*	&W	
Multitech MT1432 MT932	E0	Q1	&S1	&W	
Digicom Scout Plus	E0	Q2	&B2	&W	
Digicom SoftModem	E0	Q1	&S1	&W	
Viva 14.4/9642c	E0	Q1	&S1	&W	
ZyXel U-1496E	E0	Q1	&S1	&W	Additional information on ftp.zyxel.com
Supra V.32*bis*/28.8	E0	Q1	&S1	&W	
ZOOM 14.4	E0	Q1	&S1	&W	

Table 51 AUX and Platform Specific Settings (continued)

Modem	Settings for Use with AUX Port		Other Settings		Comments
	No Echo	No Res	CAB-MDCE	Write Memory	
Intel External	E0	Q1	*NA*	&W	
Practical Peripherals	E0	Q1	*NA*	&W	Based on PC288LCD. May vary.

Table 52 contains a legend of symbols used in Table 50 and Table 51.

Table 52 Legend to Symbols Used in Modem Chart

Symbol	Meaning
NA	This option is not available on the noted modem.
-->	The command noted on the right will handle that function.
<--	The command noted on the left will handle that function.
AUX port	These parameters are only required for pre-9.21 AUX ports or any other port without modem control set.

Sample Modem Scripts

The following are several modem command strings that are appropriate for use with your access server or router. For use with the access server, **Speed=xxxxxx** is a suggested value only. Set the DTE speed of the modem to its maximum capability. By making a reverse Telnet connection in the EXEC mode to the port on the access server where the modem is connected, then sending an **at** command followed by a carriage return.

In the following example, the modem is attached to asynchronous interface 2 on the access server. The IP address indicated as the server-ip-address is the IP address of the Ethernet 0 interface. The administrator connects from the EXEC to asynchronous interface 2, which has its IP address assigned from Ethernet 0.

```
2511> telnet server-ip-address port-number
              192.156.154.42    2002
```

AST Premium Exec Internal Data/Fax (MNP 5)

```
Init=AT&F&C1&D3\G0\J0\N3\Q2S7=60S0=1&W
Speed=9600
```

ATi 9600etc/e (V.42*bis*)

```
Init=AT&FW2&B1&C1&D3&K3&Q6&U1S7=60S0=1&W
Speed=38400
```

AT&T Paradyne KeepInTouch Card Modem (V.42*bis*)

```
Init=AT&FX6&C1&D3\N7\Q2%C1S7=60S0=1&w
Speed=57600
```

AT&T ComSphere 3800 Series (V.42*bis*)

```
Init=AT&FX6&C1&D2\N5\Q2%C1"H3S7=60S0=1&W
Speed=57600
```

AT&T DataPort Fax Modem (V.42*bis*)

```
Init=AT&FX6&C1&D2\N7\Q2%C1S7=60S0=1&W
Speed=38400
```

Boca Modem 14.4K/V.32*bis* (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5%C1\N3S7=60S36=7S46=138S95=47S0=1&W
Speed=57600
```

CALPAK MXE-9600

```
Init=AT&F&C1&D3S7=60S0=1&W
Speed=9600
```

Cardinal 2450MNP (MNP 5)

```
Init=AT&F&C1&D3\J0\N3\Q2\V1%C1S7=60S0=1&w
Speed=9600
```

Cardinal 9650V32 (MNP)

```
Init=AT&F&B1&C1&D3&H1&I1&M6S7=60S0=1&W
```

Cardinal 9600V42 (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5\N3%C1%M3S7=60S46=138S48=7S95=3S0=1&W
Speed=38400
```

Cardinal 14400 (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5\N3%C1%M3S7=60S46=138S48=7S95=47S0=1&W
Speed=57600
```

COMPAQ SpeedPAQ 144 (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5\J0\N3%C1S7=60S36=7S46=2S48=7S95=47S0=1&W
Speed=57600
```

Data Race RediMODEM V.32/V.32*bis*

```
Init=AT&F&C1&D3&K3&Q6\J0\N7\Q3\V2%C1S7=60 Speed=38400S0=1&W
```

Dell NX20 Modem/Fax (MNP)

```
Init=AT&F&C1&D3%C1\J0\N3\Q3\V1W2S7=60S0=1&W
Speed=9600
```

Digicom Systems (DSI) 9624LE/9624PC (MNP 5)

```
Init=AT&F&C1&D3*E1*F3*S1S7=60S0=1&W
```

Digicom Systems (DSI) 9624LE+ (V.42*bis*)

```
Init=AT&F&C1&D3*E9*F3*N6*S1S7=60S0=1&W
Speed=38400
```

Everex Evercom 24+ and 24E+ (MNP 5)

```
Init=AT&F&C1&D3\J0\N3\Q2\V1%C1S7=60S0=1&W
```

Everex EverFax 24/96 and 24/96E (MNP 5)

```
Init=AT&F&C1&D3\J0\N3\Q2\V1%C1S7=60S0=1&W
Speed=9600
```

Everex Evercom 96+ and 96E+ (V.42bis)

```
Init=AT&FW2&C1&D3\J0\N3\Q2\V2%C1S7=60S0=1&W
Speed=38400
```

Freedom Series V.32*bis* Data/FAX Modem

```
Init=AT&F&C1&D3&K3&Q6\J0\N7\Q3\V2%C1S7=60S0=1&W
Speed=38400
```

Gateway 2000 TelePath

```
Init=AT&FW2&C1&D3&K3&Q5\N3%C1S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=38400
```

Gateway 2000 Nomad 9600 BPS Internal Modem

```
Init=AT&F&C1&D3%C1\J0\N3\Q2S7=60S0=1&W
Speed=38400
```

GVC SM-96V (V.42*bis*)

```
Init=AT&F&C1&D3%C1\J0\N6\Q2\V1S7=60S0=1&W
Speed=38400
```

GVC SM-144V (V.42*bis*)

```
Init=AT&F&C1&D3%C1\J0\N6\Q2\V1S7=60S0=1&W
Speed=57600
```

Hayes Smartmodem Optima 9600 (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5S7=60S46=138S48=7S95=47S0=1&W
Speed=38400
```

Hayes Smartmodem Optima 14400 (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5S7=60S46=138S48=7S95=47S0=1&W
Speed=57600
```

Hayes Optima 28800 (V.34)

```
Init=AT&FS0=1&C1&D3&K3&Q6&Q5&Q9&W
Speed=115200
```

Hayes V-series Smartmodem 9600/9600B (V.42)

```
Init=AT&F&C1&D3&K3&Q5S7=60S0=1&W
Speed=9600
```

Hayes V-series ULTRA Smartmodem 9600 (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5S7=60S46=2S48=7S95=63S0=1&W
Speed=38400
```

Hayes V-series ULTRA Smartmodem 14400 (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5S7=60S38=10S46=2S48=7S95=63S0=1&W
Speed=38400
```

Hayes ACCURA 24 EC (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5S7=60S36=7S46=138S48=7S95=47S0=1&W
```

Hayes ACCURA 96 EC (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=38400
```

Hayes ACCURA 144 EC (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=57600
```

Hayes ISDN System Adapter

```
Init=AT&FW1&C1&D3&K3&Q0S7=60S0=1&W
Speed=57600
```

IBM 7855 Modem Model 10 (MNP)

```
Init=AT&F&C1&D3\N3\Q2\V1%C1S7=60S0=1&W
```

IBM Data/Fax Modem PCMCIA (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5%C3\N3S7=60S38=7S46=138S48=7S95=47S0=1&W
Speed=57600
```

Identity ID9632E

```
Init=AT&F&C1&D3S7=60S0=1&W
Speed=9600
```

Infotel V.42X (V.42*bis*)

```
Init=AT&F&C1&D3S7=30S36=7S0=1&W
Speed=9600
```

Infotel V.32 turbo (V.42*bis*)

```
Init=AT&FW1&C1&D3&K3&Q5S7=60S0=1&w
Speed=38400
```

Infotel 144I (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5\N3%C1S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=38400
```

Intel 9600 EX (V.42*bis*)

```
Init=AT&F&C1&D3\J0\N3\Q2\V2%C1"H3S7=60S0=1&W
Speed=38400
```

Intel 14400 EX (V.42*bis*)

```
Init=AT&F&C1&D3\J0\N3\Q2\V2%C1"H3S7=60S0=1&W
Speed=38400
```

Macronix MaxFax 9624LT-S

```
Init=AT&F&C1&D3&K3&Q9\J0\N3\Q3%C1S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=9600
```

Megahertz T3144 internal (V.42*bis*)

```
Init=AT&F&C1&D3%C1\J0\N3\Q2\V2S7=60S0=1&W
Speed=57600
```

Megahertz T324FM internal (V.42*bis*)

```
Init=AT&F&C1&D3%C1\J0\N3\Q2\V1S7=60S46=138S48=7S0=1&W
Speed=9600
```

Megahertz P2144 FAX/Modem (V.42*bis*)

```
Init=AT&F&C1&D3%C1\J0\N7\Q2\V2S7=60S0=1&W
Speed=38400
```

Megahertz T396FM internal (V.42*bis*)

```
Init=AT&FW2&C1&D3%C1\J0\N7\Q2\V2S7=60S0=1&W
Speed=38400
```

Megahertz CC3144 PCMCIA card modem (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5%C3\N3S7=60S38=7S46=138S48=7S95=47S0=1&W
Speed=57600
```

Microcom AX/9624c (MNP 5)

```
Init=AT&F&C1&D3\G0\J0\N3\Q2%C1S7=60S0=1&W
Speed=9600
```

Microcom AX/9600 Plus (MNP 5)

```
Init=AT&F&C1&D3\J0\N3\Q2S7=60S0=1&W
```

Microcom QX/V.32c (MNP 5)

```
Init=AT&F&C1&D3\J0%C3\N3\Q2S7=60S0=1&W
Speed=38400
```

Microcom QX/4232hs (V.42*bis*)

```
Init=AT&F&C1&D3\J0%C3\N3\Q2-K0\V2S7=60S0=1&W
Speed=38400
```

Microcom QX/4232*bis* (V.42*bis*)

```
Init=AT&F&C1&D3\J0%C3\N3\Q2-K0\V2W2S7=60S0=1&W
Speed=38400
```

Microcom Deskporte 28800 (V.34)

```
Init=AT&F&c1&q1E0S0=1&W
Speed=115200
```

Microcom MicroPorte 542 (V.42*bis*)

```
Init=AT&F&C1&D3&Q5S7=60S46=138S48=7S95=47S0=1&W
Speed=9600
```

Microcom MicroPorte 1042 (V.42*bis*)

```
Init=AT&F&C1&D3%C3\J0-M0\N6\Q2\V2S7=60S0=1&W
Speed=9600
```

Microcom MicroPorte 4232*bis* (V.42*bis*)

```
Init=AT&F&C1&D3%C3%G0\J0-M0\N6\Q2\V2S7=60S0=1&W
Speed=38400
```

Microcom DeskPorte FAST

```
Init=ATX4S7=60-M1\V4\N2L1S0=1&W
Speed=57600
```

Motorola/Codex 3220 (MNP)

```
Init=AT&F&C1&D3*DC1*FL3*MF0*SM3*XC2S7=60S0=1&W
```

Motorola/Codex 3220 Plus (V.42*bis*)

```
Init=AT&F&C1&D3*DC1*EC0*MF0*SM3*XC2S7=60S0=1&W
Speed=38400
```

Motorola/Codex 326X Series (V.42*bis*)

```
Init=AT&F&C1&D3*FL3*MF0*SM3*TT2*XC2S7=60S0=1&W
Speed=38400
```

MultiTech MultiModem V32EC (V.42*bis*)

```
Init=AT&FX4&C1&D3$BA0&E1&E4&E15#L0S7=60S0=1&W
Speed=38400
```

MultiTech MultiModem V32 (no MNP or V.42)

```
Init=AT&F&C1&D3S7=60S0=1&W
Speed=9600
```

MultiTech MultiModem 696E (MNP)

```
Init=AT&F&C1&D3$BA0&E1&E4&E15S7=60S0=1&W
```

MultiTech MultiModem II MT932 (V.42*bis*)

```
Init=AT&FX4&C1&D3$BA0&E1&E4&E15#L0S7=60S0=1&W
Speed=38400
```

MultiTech MultiModem II MT1432 (V.42*bis*)

```
Init=AT&FX4&C1&D3#A0$BA0&E1&E4&E15#L0S7=60S0=1&W
Speed=57600
```

NEC UltraLite 14.4 Data/Fax Modem (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q4\J0\N7\Q2W2%C1S7=60S0=1&W
Speed=38400
```

Practical Peripherals PC28800SA (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5S7=60S36=7S46=2S48=7S95=47S0=1&W
Speed=115200
```

Practical Peripherals PM9600SA (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5S46=138S48=7S7=60S0=1&W
Speed=38400
```

Practical Peripherals PM14400FX (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5S7=60S36=7S46=2S48=7S95=47S0=1&W
Speed=57600
```

Practical Peripherals PM14400SA (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5S7=60S36=7S46=2S48=7S95=47S0=1&W
Speed=57600
```

Prometheus ProModem 9600 Plus (V.42)

```
Init=AT&F&C1&D3*E7*F3S7=60S0=1&W
```

Prometheus ProModem Ultima (V.42*bis*)

```
Init=AT&F&C1&D3*E9*F3*N6*S1S7=60S0=1&W
Speed=38400
```

Racal Datacomm ALM 3223 (V.42*bis*)

```
Init=AT&F&C1&D3\M0\N3\P2\Q1\V1S7=60S0=1&W
Speed=38400
```

Supra FAXModem V.32*bis* (V.42*bis*)

```
Init=AT&FN1W2&C1&D1&K3&Q5\N3%C1S7=60S36=7S48=7S95=45S0=1&W
Speed=57600
```

Telebit T1600 (V.42*bis*)

```
Init=AT&FX2&C1&D3&R3S7=60S51=6S58=0S59=15S68=2S180=2S190=1S0=1&W
Speed=38400
```

Telebit T2500 (V.42*bis*)

```
Init=AT~&FX2S7=60S51=5S52=2S66=1S68=2S97=1S98=3S106=1S131=1S0=1&W

Telebit T3000 (V.42bis)
Init=AT&FX2&C1&D3S51=6S59=7S68=2S7=60S0=1&W
Speed=38400
```

Telebit QBlazer (V.42*bis*)

```
Init=AT&FX2&C1&D3S59=7S68=2S7=60S0=1&W
Speed=38400
```

Texas Instruments V.32*bis* Internal Modem

```
Init=AT&F&C1&D3%C1\J0\N7\Q2\V2S7=60S0=1&W
Speed=38400
```

Toshiba T24/DF Internal

```
Init=AT&F&C1&D3\J0\N3\Q2%C1S7=60S36=7S46=138S48=7S0=1&W
Speed=9600
```

Universal Data Systems FasTalk V.32/42b (V.42*bis*)

```
Init=AT&F&C1&D3\J0\M0\N7\V1\Q2%C1S7=60S0=1&W
Speed=38400
```

Universal Data Systems V.32 (no MNP or V.42)

```
Init=AT&F&C1&D2S7=60S0=1&W
Speed=9600
```

Universal Data Systems V.3224 (MNP 4)

```
Init=AT&F&C1&D2\J0\N3\Q2S7=60S0=1&W
```

Universal Data Systems V.3225 (MNP 5)

```
Init=AT&F&C1&D2\J0\N3\Q2%C1S7=60S0=1&W
```

Universal Data Systems V.3227 (V.42*bis*)

```
Init=AT&F&C1&D2\J0\M0\N7\Q2%C1S7=60S0=1&W
Speed=38400
```

Universal Data Systems V.3229 (V.42*bis*)

```
Init=AT&F&C1&D3\J0\M0\N7\Q2%C1S7=60S0=1&W
Speed=38400
```

US Robotics Sportster 9600 (V.42*bis*)

```
Init=AT&FX4&A3&B1&D3&H1&I0&K1&M4S7=60S0=1&W
Speed=38400
```

US Robotics Sportster 14400 (V.42*bis*)

```
Init=AT&FX4&A3&B1&D3&H1&I0&K1&M4S7=60S0=1&W
Speed=57600
```

US Robotics Sportster 14400 (V.42*bis*) x

```
Init=AT&FX4&B1&C1&D2&H1&K1&M4E0X7Q0V1S0=1&W
Speed=57600
```

US Robotics Sportster 28800 (V.34)

```
Init=AT&FS0=1&C1&D2&H1&R2&N14&B1&W
Speed=115200
```

US Robotics Courier 28800 (V.34)

```
Init=AT&FS0=1&C1&D2&H1&R2&N14&B1&W
Speed=115200
```

US Robotics Courier V.32bis (V.42*bis*)

```
Init=AT&FX4&A3&C1&D2&M4&H1&K1&B1S0=1&W
Speed=38400
```

US Robotics Courier HST Dual Standard (V.42*bis*)

```
Init=AT&FB0X4&A3&C1&D2&M4&H1&K1&B1&R2&S1S0=1&W
Speed=115200
```

US Robotics Courier HST (V.42*bis*)

```
Init=AT&FB0X4&A3&C1&D2&M1&H1&K1&B1S0=1&W
Speed=115200
```

US Robotics WorldPort 2496 FAX/Data (V.42*bis*)

```
Init=AT&FX4&C1&D3%C1"H3\J0-J1\N3\Q2\V2S7=60S0=1&W
Speed=57600
```

US Robotics WorldPort 9696 FAX/Data (MNP 5)

```
Init=AT&FX4&C1&D3%C1\J0\N3\Q2\V2S7=60S0=1&W
```

US Robotics WorldPort 9600 (MNP 5)

```
Init=AT&FX4&C1&D3%C1\J0\N3\Q2\V2S7=60S0=1&W
```

US Robotics WorldPort 14400 (V.42*bis*)

```
Init=AT&FX4&A3&B1&C1&D3&H1&K1&M4S7=60S0=1&W
Speed=57600
```

Ven-Tel PCM 9600 Plus (MNP)

```
Init=AT&FB0&C1&D3\N3\Q3%B0%C1%F1S7=60S0=1&W
```

ViVa 9642e (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5\N3%C3S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=38400
```

ViVa 14.4/FAX (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5\N3%C3S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=38400
```

ZOOM V.32 turbo (V.42*bis*)

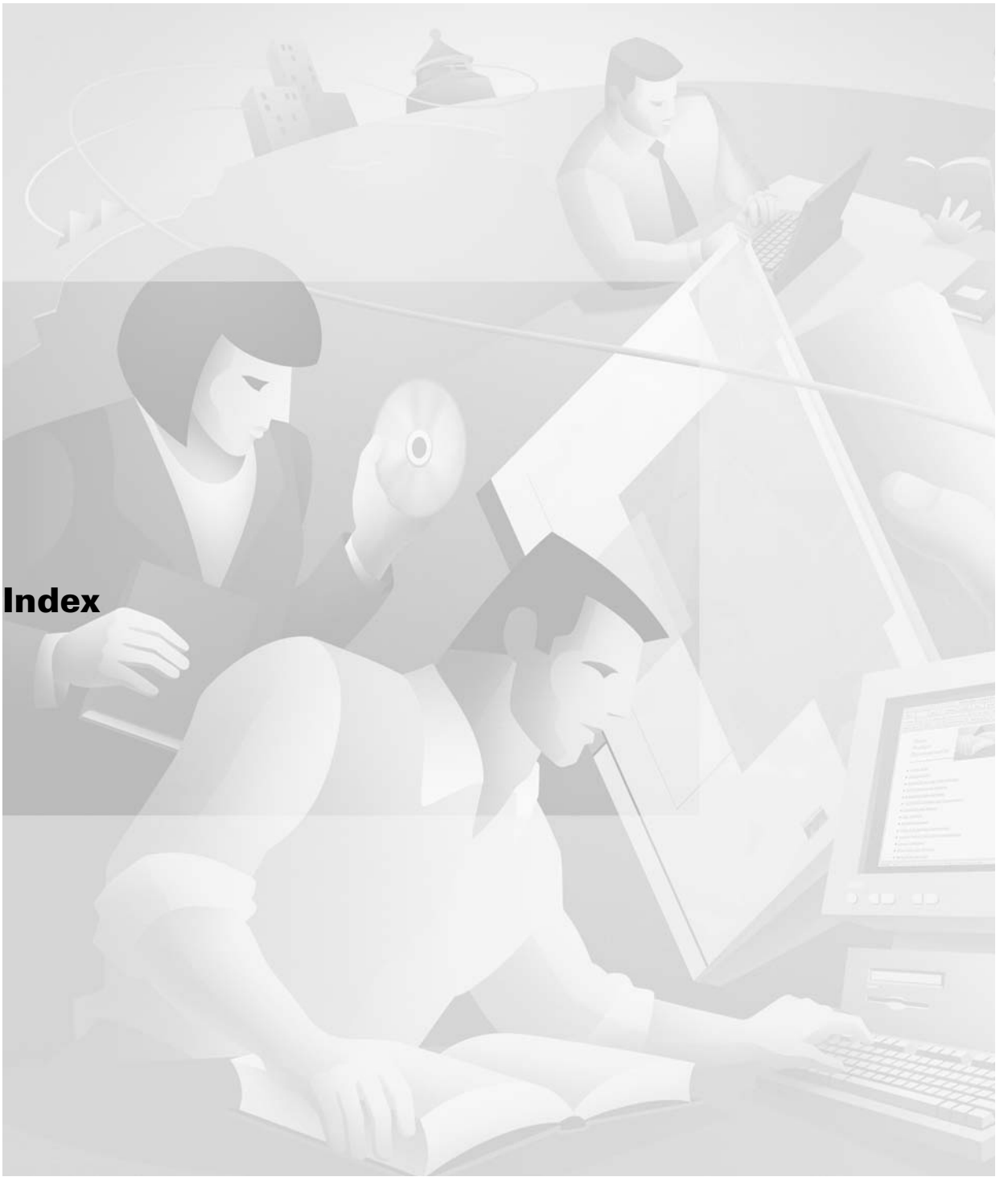
```
Init=AT&FW1&C1&D3&K3&Q5%C1\N3S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=38400
```

ZOOM V.32*bis* (V.42*bis*)

```
Init=AT&FW1&C1&D3&K3&Q9%C1\N3S7=60S36=7S95=47S0=1&W
Speed=38400
```

Zyxel U-1496 (V.42*bis*)

```
Init=AT&FX6&B1&C1&D2&N0&K4&H3S7=60S0=1&W
Speed=57600
```



Index

Symbols

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X