

# Configuring Dial-on-Demand Routing

---

This chapter describes how to configure your communication server for dial-on-demand routing (DDR) and dial backup. For a complete description of the commands mentioned in this chapter, refer to the *Communication Server Command Reference* publication. For historical background and a technical overview of DDR, see the *Internetworking Technology Overview* publication.

## Cisco's Implementation of Dial Backup and DDR

Dial backup provides protection against WAN downtime by allowing you to configure a backup serial line circuit-switched connection. Dial backup software keeps the secondary line inactive (DTR inactive) until one of the following conditions is met:

- The primary line goes down.
- The transmitted traffic load on the primary line exceeds a defined limit.

The secondary line and these conditions are defined using the **backup interface**, **backup load**, and **backup delay** interface configuration commands.

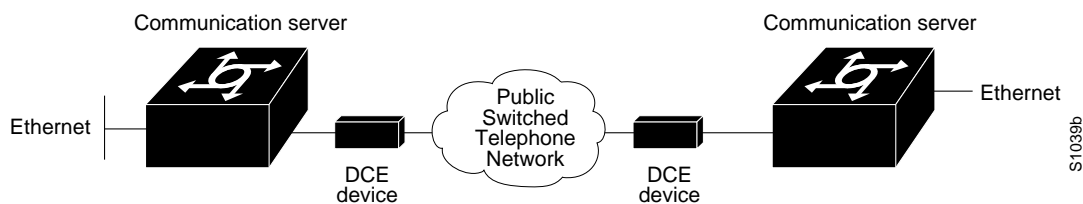
When the software detects a lost Carrier Detect signal from the primary line device or finds that the line protocol is down, it activates DTR on the secondary line. At that time, the DCE must be set to dial the remote site. When the connection is made, the routing protocol defined for the serial line will take over the task of transmitting traffic over the dialup line.

Dial-on demand routing (DDR) provides network connections across the Public Switched Telephone Network (PSTN). Traditionally, networks have been interconnected using dedicated lines for WAN connections. With DDR, you can use modems to establish low-volume, periodic network connections over public circuit-switched networks.

Synchronous and asynchronous interfaces can be configured for DDR connections to one or more destination networks. When a packet is received for a remote network, the communication server uses dialing commands to send the phone number of the destination network to a modem. The modem (DCE device) then dials the destination DCE device and establishes a connection.

Figure 1-1 illustrates a typical DDR interconnection configuration.

Figure 1-1 Dial-on-Demand Routing Interconnection



## Placing Calls Using DDR

On the communication server, DDR places calls using the following two methods:

- Chat scripts for asynchronous interfaces
- V.25bis for synchronous interfaces

## Chat Scripts on Asynchronous Interfaces

On asynchronous lines, our communication servers support chat scripts used to send commands for modem dialing and logging on to remote systems. To dial a call on an asynchronous line, a chat script must be defined. If multiple chat scripts are defined, regular expressions are used for powerful pattern matching to select between many scripts. See Appendix B, "Regular Expressions," in the *Communication Server Command Reference* for information about regular expressions.

A *chat script* is a string of text that defines the login "conversation" that occurs between two systems. It consists of expect-send pairs which define the string the local system expects to receive from the remote system and what the local system should send as a reply.

## V.25bis over Synchronous Interfaces

Our communication servers support connections from the synchronous serial interface to any DCE device that supports V.25bis. V.25bis is a CCITT recommendation for initiating calls using in-band signaling. Depending on the type of modem or CSU/DSU you are using, CCITT V.25bis options might be supported.

The V.25bis specification describes two modes of establishing or receiving calls: the *direct call mode* and the *addressed call mode*. Communication servers support connections using the addressed call mode and synchronous, bit-oriented operation. The addressed call mode allows control signals and commands to be sent over the DCE data interface to establish and terminate calls. These commands are packaged in High-level Data Link Control (HDLC) synchronous data frames.

Devices used by the communication server for dialing out must support certain hardware signals in addition to V.25bis. When the communication server drops DTR, the device must disconnect any calls that are currently connected. When the device connects to the remote end, Data Carrier Detect (DCD) must be automatically asserted.

---

**Note** For many V.25bis devices, raised DCD requires a special cable to crossover DCD and Data Set Ready (DSR) signals, because the V.25bis specification requires DSR to be raised when a connection is established.

---

Table 1-1 lists V.25bis options. The V.25bis options are supported in the dial string (telephone number) only if you have enabled DDR using the **dialer in-band** command. The functions of these options are nation-specific, and they might have different implementations in your country. Refer to your modem manual for a list of supported options.

**Table 1-1 CCITT V.25bis Options**

Option	Description
:	Wait tone.
<	Pause. Usage and duration of this parameter vary by country.
=	Separator 3. For national use.
>	Separator 4. For national use.
P	Dialing to be continued in pulse mode. Optionally accepted parameter.
T	Tone. (Dialing to be continued in Dual Tone Multifrequency, DTMF mode) Optionally accepted parameter.
&	Flash. (The flash duration varies by country.) Optionally accepted parameter.

## Controlling Access for DDR

DDR supports a variety of security and access control methods including the following:

- Dialer access lists and dialer access groups—Based on the access lists configured, access groups control access for dial-on-demand routing.

Packets that are permitted entry according to the access list are identified as “interesting” or “packets of interest.” Packets that are not permitted entry or are denied entry by an access list are deemed “uninteresting.”

A communication server activates the dial-on-demand feature when it receives an interesting packet destined for a location that can be reached over a dialed connection through a PSTN. After the communication server dials the destination phone number and establishes a connection, packets can be transmitted. When the transmission is complete, after a configured period of line time during which there is no interesting traffic on the line, the line is automatically disconnected.

---

**Note** TCP/IP routing protocols IS-IS, BGP, and OSPF are not recommended with DDR because they require an acknowledgment for routing updates. Because DDR lines are brought up as needed, DDR will not necessarily be active and available to send responses at the times the updates are sent.

---

- Address mapping—Interfaces can be configured to map a next hop address to a phone number. This allows the communication server to forward packets to the correct destinations, and to determine if a connection is already established to that destination.

- CHAP—Access control using Challenge Handshake Authentication Protocol (CHAP) can be configured on serial interfaces that have PPP encapsulation enabled. CHAP reduces the risk of security violations on your communication server. CHAP also serves a method to identify incoming calls.

---

**Note** Access lists must be defined before you can use DDR. If there are no access lists defined, access is implicitly denied. See the IP chapter in this manual for information about the IP access lists with the **tcp** keyword specified and the IPX chapter for information about the IPX access lists.

---

## Dial Backup Configuration Task List

You must first decide whether you want to backup when the primary line goes down, when traffic load on the primary line exceeds the defined threshold, or both. The tasks you perform depend on your decision. Perform one or more of the following tasks to configure dial backup:

- Select Backup Line
- Define the Traffic Load Threshold
- Define Backup Line Delays

## Select Backup Line

To configure dial backup, set a secondary serial interface as a backup to a primary serial interface. An external data communications equipment (DCE) device such as a modem, attached to a circuit-switched service must be connected to the secondary serial interface. The external device must be capable of responding to a DTR signal (DTR active) by auto-dialing a connection to a preconfigured remote site.

To select a backup line, perform the following task in interface configuration mode:

Task	Command
Select a backup line.	<b>backup interface</b> <i>interface number</i>

The interface specified in the **backup interface** command can only backup one interface. For an example of selecting a backup line, see “Example of Dial Backup” later in this chapter.

## Define the Traffic Load Threshold

You can configure dial backup to activate the secondary line based on the traffic load on the primary line. The software monitors the traffic load and computes a five-minute moving average. If this average exceeds the value you set for the line, the secondary line is activated and, depending upon how the line is configured, some or all of the traffic will flow onto the secondary dialup line.

To define how much traffic should be handled at one time on an interface, perform the following task in interface configuration mode:

Task	Command
Define the traffic load threshold as a percentage of the primary line’s available bandwidth.	<b>backup load</b> { <i>enable-threshold</i>   <b>never</b> } { <i>disable-load</i>   <b>never</b> }

## Define Backup Line Delays

You can configure a value that defines how much time should elapse before a secondary line status changes after a primary line status changed. This means you can define two delays:

- A delay that apply after the primary line goes down but before the secondary line is activated
- A delay that applies after the primary line goes up but before the secondary line is activated

To define these delays, perform the following task in interface configuration mode:

Task	Command
Define backup line delays.	<b>backup delay</b> { <i>enable-delay</i>   <b>never</b> } { <i>disable-delay</i>   <b>never</b> }

For an example of how to define backup line delays, see “Example of Dial Backup” later in this chapter.

## DDR Configuration Task Overview

Before you configure the asynchronous interface to support DDR, configure the line as follows:

- Specify line speed.
- Set flow control, if any.

- Specify your modem type.

See the “Configuring Lines and Terminal Setting,” chapter for more information about setting line speed, flow control, and modem type.

One of the following three tasks is required to configure your communication server for dial-on-demand routing:

- Configure an Interface to Place Calls
- Configure an Interface to Receive Calls
- Configure an Interface to Place and Receive Calls

The following are optional tasks to customize, enhance, and monitor DDR:

- Customize the DDR Network
- Monitor DDR Connections
- Configure DDR for IPX

The following sections describe these tasks. See the “DDR Configuration Examples” section later in this chapter for an idea of how to configure DDR on your network.

## Configure an Interface to Place Calls

The following tasks are required to configure a single interface, multiple interfaces, or dialer rotary groups to place calls:

- Step 1** Create chat scripts (for asynchronous interfaces only).
- Step 2** Specify a chat script for a line (asynchronous).
- Step 3** Configure to call a single site or multiple sites.

## Configure Chat Scripts for Asynchronous Interfaces

A chat script must be defined for dialing out on asynchronous lines, specifically the asynchronous interface. Chat scripts are used to send commands for modem dialing and logging on to remote systems.

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

Task	Command
Create a script that will place a call on a modem and/or log on to a remote system.	<b>chat-script</b> <i>script-name expect send...</i>

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

For an example of how to use chat scripts, see “Example of Using Chat Scripts” later in this chapter.

## Suggested Chat Script Naming Conventions

A suggested chat script naming convention is as follows:

**vendor-type-modulation**

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

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

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

**telebit-t3000-v32bis**

The chat-script command could become the following:

```
cs(config)# chat-script telebit-t3000-v32bis ABORT ERROR ABORT BUSY ABORT
"NO ANSWER" "" "AT H" OK "AT DT \T" DIALING \c TIMEOUT 30 CONNECT \c
```

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

## Specify Chat Scripts for Asynchronous Lines

After a chat script has been defined, it must be applied to a line or an interface before it will be used. To specify a chat script for a line, perform the following task in line configuration mode:

Task	Command
Specify a modem script for a line.	<b>modem chat-script</b> <i>regexp</i>

A maximum of one **modem chat-script** command can be configured per line. The chat script naming convention described in the previous section allows you to specify a chat script by the type of the modem attached to that line as follows:

**modem chat-script modem-type\***

UNIX-style regular expressions are used to match patterns and select between many scripts. This will be useful if you specify modem scripts on an interface that is used to dial multiple destinations. Dialing multiple destinations is described in the “Configure for Placing Calls to a Single Site” section later in this chapter. Regular expressions are described in “Regular Expressions” appendix in the *Communication Server Command Reference*.”

## Configure for Placing Calls to a Single Site

The modem chat script becomes the default chat script for an interface. That means that it becomes the default chat script for **dialer string** and **dialer map** commands, presented in this section.

Perform the following tasks to configure an interface for calling a single site:

**Step 1** Enable DDR on the interface.

**Step 2** Specify the dial string (synchronous interfaces)

Or, specify chat scripts and dial strings (asynchronous interfaces).



To call a single site, you must enable DDR using the **dialer-in-band** command. Dialers specified by this command use chat scripts on asynchronous interfaces and V.25bis on synchronous interfaces. If using V.25bis, you can optionally specify parity. The 1984 version of the V.25bis specification states that characters must have odd parity. However, the default is no parity.

To enable DDR, perform the following task in interface configuration mode:

Task	Command
Enable DDR on a serial interface. Set parity on synchronous serial interfaces only.	<b>dialer in-band</b> [ <b>no-parity</b>   <b>odd-parity</b> ]

If you want to call only one remote system per interface, the **dialer string** command is useful. You do not need to use the **dialer map** command for authentication. Dialers pass the string you have defined to the external DCE.

To specify the string (telephone number) to be called on serial interfaces (asynchronous or synchronous), perform the following task in interface configuration mode:

Task	Command
Specify a string of numbers to dial.	<b>dialer string</b> <i>dial-string</i>

**Note** For asynchronous interfaces that do not require a system script, a modem script must be defined for the associated line by using the **modem chat-script** line configuration command.

To place a call to a single site on an asynchronous line for which a modem script has not been assigned and/or a system script must be specified, perform the following task in interface configuration mode.

Task	Command
Specify chat scripts and a dial string.	<b>dialer map</b> <i>protocol next-hop-address</i> [ <b>modem-script</b> <i>modem-regexp</i> ] [ <b>system-script</b> <i>system-regexp</i> ] <i>dial-string</i>

Use the **dialer-map** command to specify a chat script if the following is true:

- No modem script is specified for the line and/or an additional (system) chat script is required to log onto the remote system.

You do not need to specify a system script if the following is true:

- The modem script can be used to dial and log on to the remote system
- Or, you are calling a system that does not require a login script; that is, a system that answers and immediately goes into protocol mode.

## Configure for Placing Calls to Multiple Sites

You can configure your communication server to call multiple sites from a single line, from multiple lines, or from a dialer rotary group.

### Call on a Single Line or Multiple Lines

To configure your communication server to call multiple sites on a single line or on multiple lines, perform the following tasks:

**Step 1** Enable DDR on the interface.

**Step 2** Define multiple dialing destinations on the interface.

Or, specify a string of numbers to dial.

To enable DDR on an interface, perform the following task in interface configuration mode:

Task	Command
Enable DDR on a serial interface. Set parity on synchronous serial interfaces only.	<b>dialer in-band</b> [no-parity   odd-parity]

To define dialing destinations, perform one of the following tasks:

Task	Command
Define multiple dialing destinations on a synchronous interface.	<b>dialer map</b> protocol next-hop-address dial-string
Define multiple dialing destinations on an asynchronous interface.	<b>dialer map</b> protocol next-hop-address [ <b>modem-script</b> modem-regexp] [ <b>system-script</b> system-regexp] dial-string
Specify a string of numbers to dial (to configure one phone number on multiple lines only.)	<b>dialer string</b> dial-string

If you adhered to the chat script naming convention described earlier in this chapter, use the form [**modem-script** \*modulation-type] in the **dialer map** command, as in “.\*-v32bis.” This allows you to specify the modulation type that is best for the system you are calling, and allows the modem type for the line to be specified by the **modem chat-script** command.

The expression “.” is a wildcard that matches any character, and the expression “\*” indicates that the preceding character can be duplicated multiple times. For more information about regular expressions, see Appendix B, “Regular Expressions,” in the *Communication Server Command Reference*.

If there is a **modem script** specified in the interface configuration command (**dialer map**) and a modem script specified in the line configuration command (**modem chat-script**), the first chat script that matches both will be used. If no script matches both, an error message is logged and the connection is not established. If there is no modem chat script specified for the line, the first chat

script (**chat-script** global configuration command) that matches the modem script regular expression will be used. If there is a system script specified in the interface configuration command, the first chat script to match the regular expression will be used.

Configure a **dialer map** command for each remote destination for that interface.

## Call on a Dialer Rotary Group

To configure your communication server to place multiple calls using a dialer rotary group, perform the following tasks:

- Step 1** Define a rotary group.
- Step 2** Enable DDR on the rotary interface.
- Step 3** Define multiple dialing destinations for the rotary group.
- Step 4** Assign physical interfaces to the rotary group.

Dialer rotary groups allow you to apply a single interface configuration to a set of physical interfaces. Dialer rotary groups are useful in environments that have multiple callers and calling destinations. Configure a dialer interface unless you are only using a single line for dialing out or have a single line dedicated to each destination.

---

**Note** The dialer rotary groups discussed in this chapter are on the communication server. The telephone company also has rotary groups that allow you to dial one rotary phone number and get connected to one of several different phone numbers. If you are using telephone company rotary groups, it is a good idea to configure dialer rotary groups on the communication server.

---

A dialer rotary group is defined by specifying a “dialer interface.” The dialer interface is not a physical interface; it is an entity that allows you to propagate an interface configuration to multiple interfaces. After the dialer interface is defined by a number, interface parameters are configured for the dialer interface. Finally, physical interfaces are assigned to the dialer rotary group. Physical interfaces inherit the interface dialer configuration parameters.

After an interface configuration is propagated to a set of physical interfaces, those interfaces can be used to place calls using standard DDR criteria. When many destinations are configured, any of the physical interfaces in a rotary group can be used for outgoing calls. When traffic arrives, an interface

from the rotary group is dialed. When more traffic for a different host arrives, another interface is dialed. Using the dialer interface allows you to specify one set of dialer maps that can apply to multiple physical lines.

You can define none to as many as nine dialer interfaces. Perform the tasks in this section for each dialer rotary group.

To define a rotary group, perform the following task in global configuration mode:

Task	Command
Define a rotary group.	<b>interface dialer</b> <i>number</i>

To enable DDR for the dialer rotary group, perform the following task in interface configuration mode:

Task	Command
Enable DDR on a serial interface. Set parity on synchronous serial interfaces only.	<b>dialer in-band</b> [ <b>no-parity</b>   <b>odd-parity</b> ]

To define multiple dialing destinations for the dialer rotary group, perform one of the following tasks in interface configuration mode:

Task	Command
Define multiple dialing destinations on a synchronous interface.	<b>dialer map</b> <i>protocol next-hop-address dial-string</i>
Define multiple dialing destinations on an asynchronous interface.	<b>dialer map</b> <i>protocol next-hop-address</i> [ <b>modem-script</b> <i>modem-regexp</i> ] [ <b>system-script</b> <i>system-regexp</i> ] <i>dial-string</i>

To assign a physical interface to a dialer rotary group, perform the following task in interface configuration mode:

Task	Command
Include the specified physical interface in a dialer rotary group in interface configuration mode.	<b>dialer rotary-group</b> <i>number</i>

Interfaces in a dialer rotary group do not have individual addresses; when the interface is being used for dialing, it inherits the parameters configured for the dialer interface. However, if the individual interface is configured with an address and it is subsequently used to establish a connection from the user EXEC level, the individual interface address again applies.

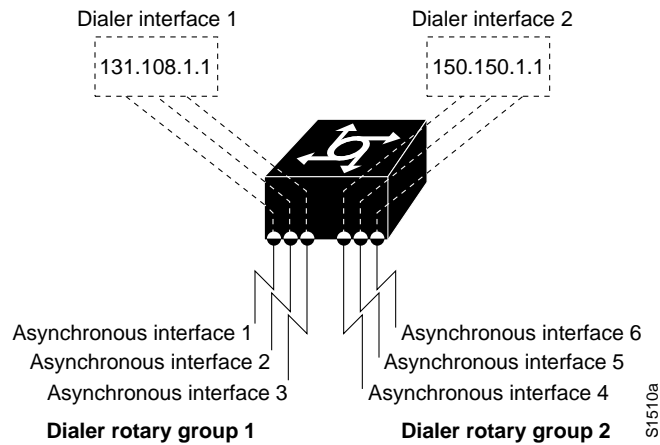
---

**Note** When you look at your configuration file, commands will not appear in the order in which you entered them. You will also see interface configuration commands that you did not enter, because interfaces inherit the parameters of the dialer interface in the dialer rotary group to which each interface has been assigned.

---

Figure 1-2 illustrates how dialer interfaces work. In this example configuration, serial interfaces 1, 2, and 3 are assigned to dialer rotary group 1. That means that these three interfaces take on the parameters configured for dialer interface 1. For example, when it is being used for dialing, the IP address of serial interface 2 is the same as the address of the dialer interface, 131.108.1.1.

Figure 1-2 Sample Dialer Interface Configuration



## Configure an Interface to Receive Calls

You can configure an interface or dialer rotary group to receive calls from a single site or from multiple sites. To configure a single line or multiple lines to receive calls from a single or multiple sites, simply enable DDR. To receive calls from multiple sites on a dialer rotary group, configure the dialer rotary group to authenticate the caller.

Perform one of the following tasks to configure an interface to receive calls:

- Configure to receive calls from a single site
- Configure to receive calls from multiple sites

---

**Note** CHAP is required for caller identification on dialer rotary groups receiving calls from multiple sites, and is described later in this section. CHAP can also be used for authentication only, in which case, an accompanying **dialer map** command is not required.

---

## Receive Calls from a Single Site

To receive a call from a single site, you must enable DDR using the **dialer-in-band** command. Dialers specified by this command use chat scripts for asynchronous interfaces and V.25bis on synchronous interfaces. Parity is not needed to enable DDR to receive calls only.

To enable DDR, perform the following task in interface configuration mode:

Task	Command
Enable DDR on a serial interface.	<b>dialer in-band [no-parity   odd-parity]</b>

## Receive Calls from Multiple Sites

You can configure your communication server to receive call from multiple sites on a single line, on multiple lines, or on a dialer rotary group.

### Receive Calls on a Single Line or Multiple Lines

No special configuration is required to receive calls on individual lines.

### Receive Calls on a Dialer Rotary Group

The following steps describe how to configure your communication server to receive calls on a dialer rotary group:

- Step 1** Assign a rotary group leader.
- Step 2** Enable DDR on the rotary interface.
- Step 3** Enable CHAP and configure CHAP authentication.
- Step 4** Assign physical interfaces to dialer rotary groups.

#### Assign a Rotary Group Leader

Dialer rotary groups allow you to apply a single interface configuration to a set of physical interfaces. Dialer rotary groups are useful in environments that have multiple callers and calling destinations. Configure a dialer interface unless you are only using a single line for dialing out.

A dialer rotary group is defined by specifying a “dialer interface.” The dialer interface is not a physical interface; it is an entity that allows you to propagate an interface configuration to multiple interfaces. After the dialer interface is defined by a number, interface parameters are configured for the dialer interface. Finally, physical interfaces are assigned to the dialer rotary group. Physical interfaces inherit the interface dialer configuration parameters.

After an interface configuration is propagated to a set of physical interfaces, those interfaces can be used to place calls using standard DDR criteria. When many destinations are configured, any of the physical interfaces in a rotary group can be used for outgoing calls. When traffic arrives, an interface

from the rotary group is dialed. When more traffic for a different host arrives, another interface is dialed. Using the dialer interface allows you to specify one set of dialer maps that can apply to multiple physical lines.

---

**Note** The dialer rotary groups discussed in this chapter are on the communication server. The telephone company also has rotary groups that allow you to dial one rotary phone number and get connected to one of several different phone numbers.

---

You can define 0 to as many as 9 dialer interfaces. For each dialer rotary group, perform the following task in global configuration mode:

Task	Command
Define a rotary group.	<b>interface dialer</b> <i>number</i>

### Enable DDR

To receive a call from multiple sites, you must enable DDR using the **dialer-in-band** command. Dialers specified by this command use chat scripts for asynchronous interfaces and V.25bis on synchronous interfaces. Parity is not needed to enable DDR to receive calls only.

To enable DDR, perform the following task in interface configuration mode:

Task	Command
Enable DDR on a serial interface.	<b>dialer in-band</b> [no parity   odd parity]

### Configure Challenge Handshake Authentication Protocol (CHAP)

Point-to-Point Protocol (PPP) with Challenge Handshake Authentication Protocol (CHAP) authentication is often used to inform the central site about which remote communication servers are connected to it.

With this authentication information, if another packet is received for a destination to which the communication server is already connected, an additional call will not be placed. However, if using rotaries, the packet will be sent out the correct port.

CHAP is specified in RFC 1333. CHAP is supported on synchronous and asynchronous serial interfaces. Using CHAP authentication, each communication server identifies itself by a *name*, which informs the other communication server what communication servers are currently connected to it. This identification process prevents a communication server from placing a call to another communication server if it is already connected to that communication server and prevents unauthorized access.

---

**Note** To use CHAP, you must be running PPP encapsulation.

---

When CHAP is enabled, a remote device (a PC, workstation, or communication server) attempting to connect to the local communication server is requested, or *challenged*, to respond. The challenge consists of a random number and the host name of the local communication server. This challenge is transmitted to the remote device. The required response is an encrypted version of a secret password, or *secret*, plus a random value and the name of the remote device.

The remote device finds the secret by looking up the host name that was received in the challenge. When the local communication server receives the challenge response, it verifies the response by looking up the name of the remote device given in the response. The secret passwords must be identical on the remote device and the local communication server. These names and secret passwords are configured using the **username** command.

By transmitting this response, the secret is never transmitted, preventing other devices from stealing it and gaining illegal access to the system. Without the proper response, the remote device cannot connect to the local communication server.

CHAP transactions occur only at the time a link is established. The local communication server does not issue a challenge during the rest of the call. (The local communication server can, however, respond to such requests from other devices during a call.)

To use CHAP, you must perform the following tasks:

- Enable PPP encapsulation.
- Enable CHAP on the interface.
- Configure host name authentication and the secret, or password, for each remote system with which authentication is required.



To enable PPP encapsulation, perform the following task in interface configuration mode:

Task	Command
Enable PPP on an interface.	<b>encapsulation ppp</b>

To enable CHAP on an interface configured for PPP encapsulation, perform the following task in interface configuration mode:

Task	Command
Enable CHAP on an interface.	<b>ppp authentication chap</b>

After you have enabled CHAP, the local communication server requires authentication from remote devices that are calling in. If the remote device does not support CHAP, no traffic will be passed to that device.

To specify the password to be used in CHAP caller identification, perform the following task in global configuration mode:

Task	Command
Configure authentication	<b>username <i>name</i> password <i>secret</i></b>

Add a **name** entry for each remote system from which the local communication server requires authentication.

The host name of each site calling in to the local communication server needs to be mapped to its address. To map a next hop to a host name, perform the following task in interface configuration mode:

Task	Command
Configure a serial interface to map host names to next hop addresses. For rotary groups only.	<b>dialer map <i>protocol next-hop-address name hostname</i></b>

### Assign an Interface to a Dialer Rotary Group

To assign a physical interface to a dialer rotary group, perform the following task in interface configuration mode:

Task	Command
Then, include the specified physical interface in a dialer rotary group in interface configuration mode.	<b>dialer rotary-group <i>number</i></b>

Interfaces in a dialer rotary group do not have individual addresses; when the interface is being used for dialing, it inherits the parameters configured for the dialer interface. However, if the individual interface is configured with an address and it is subsequently used to establish a connection from the user EXEC level, the individual interface address again applies.

## Configure an Interface to Place and Receive Calls

You can configure an interface or dialer rotary group to both place and receive calls. If the interface is calling and being called by a single site, simply enable DDR and specify a dialer string. For calling and receiving calls from multiple sites, an interface or dialer rotary group must be configured to authenticate callers and to map next hop addresses to phone numbers, or dial strings.

Perform one of the following tasks to configure an interface to place and receive calls:

- Place and receive calls from a single site
- Place and receive calls from multiple sites

### Place and Receive Calls from a Single Site

To configure your communication server to place calls to, and receive calls from, a single site, perform the following tasks in interface configuration mode:

**Step 1** Enable DDR.

**Step 2** Specify the phone number to dial.

When a dialer string is configured on an interface, and CHAP is not, any incoming call is assumed to be from the configured dialer string.

To call and receive a call from a single site, you must enable DDR using the **dialer-in-band** command. Dialers specified by this command use chat scripts on asynchronous interfaces and V.25bis on synchronous interfaces. If using V.25bis, you can optionally specify parity. The 1984 version of the V.25bis specification states that characters must have odd parity. However, the default is no parity.

To enable DDR, perform the following task in interface configuration mode:

Task	Command
Enable DDR on a serial interface. Set parity on synchronous serial interfaces only.	<b>dialer in-band [no-parity   odd-parity]</b>

To specify a dial-string destination for an interface, perform the following task in interface configuration mode:

Task	Command
Specify a string of numbers to dial.	<b>dialer string <i>dial-string</i></b>

**Note** Any incoming calls are assumed to be from the dialer string.

---

## Place and Receive Calls from Multiple Sites

To configure a single line, multiple lines, or a rotary group to place calls to and receive calls from multiple sites, perform the following tasks in interface configuration mode:

- Step 1** Enable DDR.
- Step 2** Specify a phone number to dial.
- Step 3** Configure PPP encapsulation and enable CHAP.
- Step 4** Map next hop to host name and phone number.

To call and receive calls from multiple sites, you must enable DDR using the **dialer-in-band** command. Dialers specified by this command use chat scripts on asynchronous interfaces and V.25bis on synchronous interfaces. If using V.25bis, you can optionally specify parity. The 1984 version of the V.25bis specification states that characters must have odd parity. However, the default is no parity.

To enable DDR, perform the following task in interface configuration mode:

Task	Command
Enable DDR on a serial interface. Set parity on synchronous serial interfaces only.	<b>dialer in-band [no-parity   odd-parity]</b>

To specify one destination dial-string per interface, perform the following task in interface configuration mode:

Task	Command
Specify a string of numbers to dial (to configure one phone number on multiple lines only).	<b>dialer string</b> <i>dial-string</i>

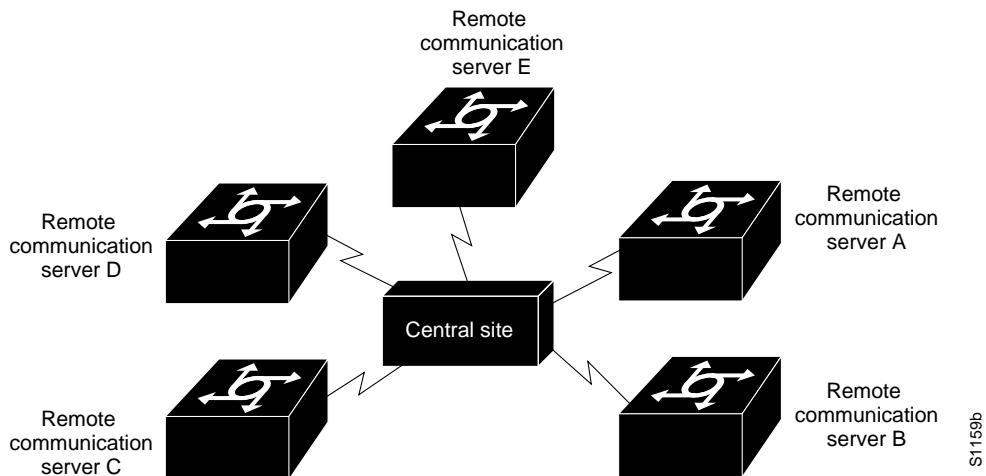
Calls from the multiple sites will need to be authenticated. Authentication can be done through CHAP. To enable CHAP on an interface and authenticate sites that are calling in, perform the following tasks in interface configuration mode:

Task	Command
Configure an interface for PPP encapsulation.	<b>encapsulation ppp</b>
Enable CHAP. <sup>1</sup>	<b>ppp authentication chap</b>
Map the next hop address to the host name and phone number.	<b>dialer map</b> <i>protocol next-hop-address</i> [ <b>modem-script</b> <i>modem-regexp</i> ] [ <b>system-script</b> <i>system-regexp</i> ] <b>name</b> <i>hostname dial-string</i>

1. For more information about configuring CHAP, see the section “Configure Challenge Handshake Authentication Protocol (CHAP)” earlier in this chapter.

Figure 1-3 shows a configuration in which the central site is calling and receiving calls from multiple sites. In this configuration, multiple sites are calling in to a central site, and the central site is calling out to the remote sites.

**Figure 1-3 Sample Configuration Using Dial-on-Demand Routing**



### Configure DDR for IPX

On DDR links for IPX, the link might come up often even when all client sessions are idle due to watchdog/keepalive packets coming from the server to all the clients approximately every five minutes. A local communication server with the DDR link can be configured to idle out the link and still make the server believe the clients are active by responding to the watchdog packets on behalf of the clients. To do so, perform the following tasks in interface configuration mode:

**Step 1** Enable DDR.

**Step 2** Enable watchdog spoofing.

You must enable DDR using the **dialer-in-band** command. Dialers specified by this command use chat scripts on asynchronous interfaces and V.25bis on synchronous interfaces. If using V.25bis, you can optionally specify parity. The 1984 version of the V.25bis specification states that characters must have odd parity. However, the default is no parity. To enable DDR, perform the following task in interface configuration mode:

Task	Command
Enable DDR	<b>dialer in-band [no parity  odd-parity]</b>

After enabling DDR, you can enable watchdog spoofing. To do so, perform the following task in interface configuration mode:

Task	Command
Enable watchdog spoofing.	<b>ipx watch-dog spoof</b>

See the “Novell IPX Commands” chapter in the *Communication Server Command Reference* for information about IPX commands.

## Customize the DDR Network

Perform the following tasks, as necessary, to customize DDR in your network.

- Set the line idle time before a line disconnects
- Set the line idle time before a line disconnects when multiple calls are going out on an interface
- Set the line down time before it can be available again
- Specify the carrier wait time
- Assign access lists to specify permit and deny conditions for DDR call access
- Specify dialer interface priority
- Set the load threshold for a dialer interface

## Set Line Idle Time

To specify the amount of time a line will stay idle before it is disconnected, perform the following task in interface configuration mode:

Task	Command
Set line idle time.	<b>dialer idle-timeout</b> <i>number-of-seconds</i>

## Set Idle Time for Busy Interfaces

The fast idle timer is activated if there is contention for a line. In other words, if a line is in use and a packet for a different next hop address is received, and the busy line is required to send the competing packet, the dialer fast idle timer is activated.

If the line has been idle for the configured amount of time, the current call is disconnected immediately and the new call is placed.

If the line has not yet been idle as long as the fast idle timer, the packet is dropped because there is no way to get through to the destination. (After the packet is dropped, the fast idle timer remains active and the current call is disconnected as soon as it has been idle for as long as the fast idle timeout.

If, in the meantime, there is another packet transmitted to the currently connected destination, and it is classified as interesting, the fast idle timer will be restarted.

Specify the amount of time a line for which there is contention will stay idle before the line is disconnected and the competing call is placed. Perform the following task in interface configuration mode:

Task	Command
Set idle time for high traffic lines.	<b>dialer fast-idle</b> <i>number-of-seconds</i>

This task applies to inbound and outbound calls.

## Set Line Down Time

To set the length of time an interface stays down before it is available to dial again after a line is disconnected or fails, perform the following task in interface configuration mode:

Task	Command
Set the interface down time.	<b>dialer enable-timeout</b> <i>number-of-seconds</i>

This task applies to inbound and outbound calls.

## Set Carrier Wait Time

To set how long to wait for the telephone service (carrier), perform the following task in interface configuration mode:

Task	Command
Set how long to wait for the carrier to come up when a call is placed.	<b>dialer wait-for-carrier-time</b> <i>number-of-seconds</i>

For asynchronous interfaces, this task sets the total time to wait for a call to connect. This time is set to allow for running the chat script.

## Controlling Access to a DDR Interface

Access lists are central to the operation of DDR. In general, access lists are used as the screening criteria for determining when to initiate DDR calls. All packets are tested against the dialer access list. Packets that match a permit entry are deemed *interesting* or *packets of interest*. Packets that do not match a permit entry or that do match a deny entry are deemed *uninteresting*. When a packet is found to be interesting, either the idle timer is reset (if the line is active) or a connection is attempted (assuming the line available but not active). If a tested packet is deemed *uninteresting*, it will be forwarded if it is intended for a destination known to be on a specific interface and the link is active. However, such a packet will not initiate a DDR call and will not reset the idle timer.

Before you perform the tasks outlined in this section, see the “Configuring IP” chapter in this manual for information about defining IP access lists and see the “Configuring Novell IPX” chapter for information about defining IPX access lists. Define access lists if you want to control access on DDR interfaces by access lists rather than protocols.

**Step 1** Associate a protocol or access list with the dialer access group.

**Step 2** Set a dialer access group number.



You can permit or deny access to an entire protocol or you can specify an access list for more refined control. Perform one of the following tasks in global configuration mode:

Task	Command
Associate an IP access list number with the dialer access group,	<b>dialer-list</b> <i>dialer-group</i> <b>list</b> <i>list-number</i>
Permit or deny a specific protocol for the dialer access group.	<b>dialer-list</b> <i>dialer-group</i> <b>protocol</b> <i>protocol-name</i> { <b>permit</b>   <b>deny</b> }

Acceptable access list numbers are IP standard and extended access list numbers, Novell IPX standard and extended access list numbers, Novell SAP access list numbers and bridge type codes. Packets matching permit conditions are deemed interesting. Packets matching deny conditions or not matching any dialer list statements are deemed uninteresting.

An interface can be associated only with a single dialer access group; multiple dialer access group assignments are not allowed. To specify the dialer access group to which you want to assign an access list, perform the following task in interface configuration mode:

Task	Command
Specify the number of the dialer access group to which the specific interface belongs.	<b>dialer-group</b> <i>group-number</i>

## Set Dialer Interface Priority

You can assign dialer priority to an interface. Priority indicates which interface in a dialer rotary group will get used first. Perform the following task in interface configuration mode.

Task	Command
Specify which dialer interfaces will be used first.	<b>dialer priority</b> <i>n</i>

For example, you might give one interface in a dialer rotary group higher priority than another if it is attached to faster, more reliable modem. In this way, the higher-priority interface will be used as often as possible.

The range of values for *n* is 0 through 255. Zero is the default value and lowest priority; 255 is the highest priority. This command applies to outgoing calls only.

## Set the Load Threshold for a Dialer Interface

You can configure a dialer rotary group to place additional calls to a single destination if the load for the interface exceeds a specified weighted value. Parallel communication server links are established based on traffic load. The number of parallel links that can be established to one location is not limited.

Task	Command
Configure the dialer rotary group to place additional calls to a single destination, as indicated by interface load.	<b>dialer load-threshold</b> <i>load</i>

## Monitor DDR Connections

Perform the following task in privileged EXEC mode to monitor DDR connections.

Task	Command
Display general diagnostics about the DDR interface.	<b>show dialer</b> [ <b>interface</b> <i>interface type</i> ]
Display status about the IPX interface.	<b>show ipx interface</b> [ <i>interface type</i> ]
Display information about the IPX packets transmitted by the communication server, including watchdog counters.	<b>show ipx traffic</b>

For information displayed by the **show** commands, refer to the *Communication Server Command Reference* publication. For more information about ipx commands, see the “Novell IPX Commands” chapter in the *Communication Server Command Reference*.

## DDR Configuration Examples

The examples provided in this section show various DDR configurations as follows:

- Example of Dial Backup
- Example of Configuring DDR in an IP Environment
- Example of Configuring DDR in an IPX Environment
- Example of Configuring Multiple Destination Dial Strings
- Example of Configuring Dialer Rotary Groups
- Example of Dialing a Single Site or Multiple Sites
- Example of Using Chat Scripts
- Example of Writing and Implementing Chat Scripts
- Example of Chat Scripts and Dialer Mapping
- Example of System Scripts and Modem Scripts
- Dial-on-Demand PPP Configuration Examples

## Example of Dial Backup

The following is an example of dial backup on async1:

```
interface serial 0
ip address 1.2.3.4 255.255.255.0
backup interface async1
backup delay 10 10
!
interface async 1
ip address 1.2.3.5 255.255.255.0
dialer in-band
dialer string 5551212
dialer-group 1
async dynamic routing
!
Dialer-list 1 protocol ip permit
!
chat-script foo "" "atdt 5551212" TIMEOUT 60 "CONNECT"
!
line aux 0
mode chat-script foo
modem inout
speed 9600
```

## Example of Configuring DDR in an IP Environment

The following example illustrates how DDR can be used on an asynchronous interface in an IP environment. The same configuration could be used on an asynchronous serial interface by changing “interface serial 1” to specify an asynchronous interface (for example, “interface async 0”).

```
interface serial 1
ip address 131.108.126.1 255.255.255.0
dialer in-band
! The next command sets the dialer idle time-out to 10 minutes
dialer idle-timeout 600
! The next command inserts the phone number
dialer string 555-1234
! The next command gives the modem enough time to recognize that
! DTR has dropped so the modem disconnects the call
pulse-time 1
! The next command adds this interface to the dialer access group defined with
! the dialer-list command
dialer-group 1
!
! The first access list statement, below, specifies that IGRP updates are not
! interesting packets. The second access-list statement specifies that all
! other IP traffic such as Ping, Telnet, or any other IP packet are interesting
! packets. The dialer-list command then creates dialer access group 1 and
! states that access list 101 is to be used to classify packets as interesting
! or uninteresting. The ip route commands
! specify that there is a route to network 131.108.29.0 and to network
! 131.108.1.0 via 131.108.126.2. This means that several destination networks
! are available through a communication server that is dialed from interface async 1.
!
access-list 101 deny igmp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
ip route 131.108.29.0 131.108.126.2
ip route 131.108.1.0 131.108.126.2
```

With many modems, the **pulse-time** command must be used so that DTR is dropped for sufficient time to allow the modem to disconnect.

The **redistribute static** command can be used to advertise static route information for DDR applications. See the **redistribute static ip** command, described in the “IP Commands” chapter in the *Communication Server Command Reference* publication. Without this command, static routes to the hosts or network that the communication server can access with DDR will not be advertised to other communication servers with which the communication server is communicating. This can block communication because some routes will not be known.

## Example of Configuring DDR in an IPX Environment

The following example illustrates how DDR can be in an IPX environment to allow the local communication server with the DDR link to idle out the link and still make the server believe the clients are active by responding to the watchdog packets on behalf of the clients. The same configuration could be used on an asynchronous serial interface by changing “interface ser 1” to specify an asynchronous interface (for example, “interface async 0”).

```
bottomright(config)# int ser 1
bottomright(config-if)# dialer in-band
bottomright(config)# no ipx route cache
bottomright(config-if)# ipx watchdog-spoof
```

## Example of Configuring Multiple Destination Dial Strings

The following example demonstrates how to specify multiple destination dial strings (phone numbers):

```
interface serial 1
ip address 131.108.126.1 255.255.255.0
dialer in-band
dialer wait-for-carrier-time 100
pulse-time 1
dialer-group 1
dialer map ip 131.108.126.10 555-8899
dialer map ip 131.108.126.15 555-5555
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 LIST 101
```

A pulse time is assigned, and a dialer access group is specified.

The first **dialer map** command specifies that the number 555-8899 is to be dialed for IP packets with a *next-hop-address* of 131.108.126.10. The second **dialer map** then specifies that the number 555-5555 will be called when an IP packet with a *next-hop-address* of 131.108.126.15 is detected.

## Example of Configuring Dialer Rotary Groups

The following configuration places interfaces serial 1 and 2 into dialer rotary group 1, defined by the **interface dialer 1** command.

```
! PPP encapsulation is enabled for interface dialer 1.
interface dialer 1
encapsulation ppp
dialer in-band
ip address 131.108.2.1 255.255.255.0
ip address 131.126.2.1 255.255.255.0 secondary

! The first dialer map command allows remote site YYY and the
! central site to call each other. The second dialer map command, with no
! dialer string, allows remote site ZZZ to call the central site but
! the central site can not call remote site ZZZ (no phone number).
dialer map ip 131.108.2.5 name YYY 1415553434
dialer map ip 131.126.2.55 name ZZZ

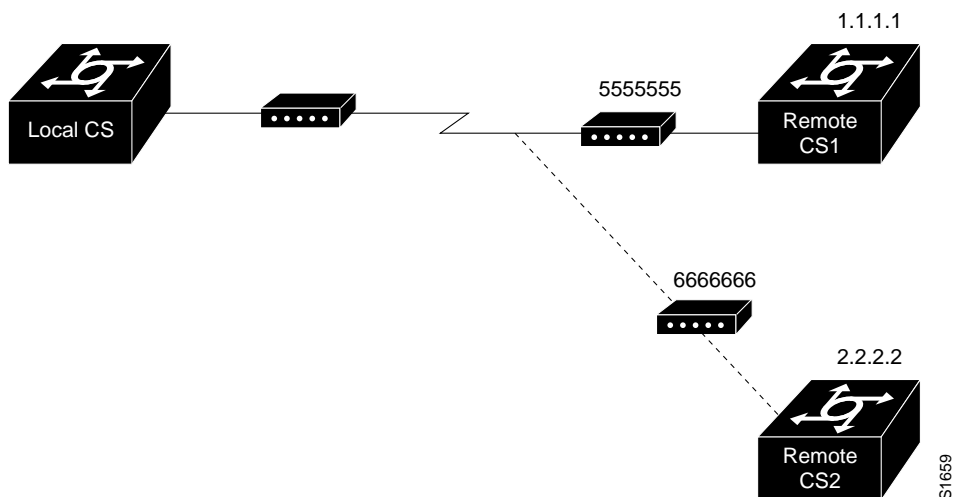
! The DTR pulse signals for three seconds on the interfaces in dialer
! group 1. This holds the DTR low so the modem can recognize that DTR has been
! dropped.
pulse-time 3

! Interfaces serial 1 and 2 are placed in dialer rotary group 1. All of
! the interface configuration commands (the encapsulation and dialer map commands shown
! earlier in this example) applied to interface dialer 1 apply
! to these interfaces.
interface serial 1
dialer rotary-group 1
interface serial 2
dialer rotary-group 1
```

## Example of Dialing a Single Site or Multiple Sites

Assume your configuration is as shown in Figure 1-4 and your communication server receives a packet with a next hop address of 1.1.1.1:

Figure 1-4 Sample Dialer String or Dialer Map Configuration



If the interface on your communication server is configured to call a single site with phone number 5555555, it will send the packet to that site, assuming that the next hop address 1.1.1.1 indicates the same remote device as phone number 5555555. The **dialer string** command is used to specify the string (telephone number) to be called.

```
interface serial 1
dialer in-band
dialer string 5555555
```

If the interface is configured to dial multiple sites, the interface or dialer rotary group must be configured so that the correct phone number, 5555555, is mapped to the address 1.1.1.1. If this mapping is not configured, the interface or dialer rotary group would not know what phone number to call to deliver the packet to its correct destination, which is the address 1.1.1.1. In this way, a packet with a destination of 2.2.2.2 will not be sent to 5555555. The **dialer map** command is used to map next hop addresses to phone numbers.

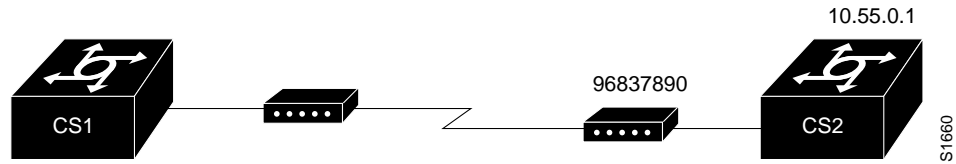
```
interface serial 1
dialer in-band
dialer map ip 1.1.1.1 5555555
dialer map ip 2.2.2.2 6666666
```

## Example of Using Chat Scripts

Figure 1-5 shows the following configuration:

- Configuration is on Communication Server A
- Chat script “dial” is used to dial Communication Server B’s modem.
- Chat script “login” is used to log in to Communication Server B.
- The phone number is the number of the modem attached to Communication Server B.
- The IP address is the address of Communication Server B.

Figure 1-5 Chat Script Configuration and Function



```

chat-script dial ABORT ERROR "" "AT Z" OK "ATDT \T" TIMEOUT 30 CONNECT \c
chat-script login ABORT invalid TIMEOUT 15 name: billw word: wewpass ">"
"slip default"
interface async 10
dialer in-band
dialer map ip 10.55.0.1 modem-script dial system-script
login 96837890

```

## Example of Writing and Implementing Chat Scripts

In the following example chat script, “ ” means expect anything and \r means send a return.

```
" " \r "name:" "myname" "ord":" "mypassword" ">" "slip default"
```

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 both the modem script “.\*-v32” and the system script “cisco.” If there is no match for both the modem script and 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 1.0.0.1 modem-script .*-v32 system-script cisco 1234

```

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
modem chat-script telebit.*
! Some lines have US robotics modems
line 7 12
modem chat-script usr.*

```

## Example of Chat Scripts and Dialer Mapping

The following example shows a dialing chat script and a login chat script. 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 login scripts.

```

chat-script dial ABORT ERROR "" "AT Z" OK "ATDT \T" TIMEOUT 30 CONNECT \c
chat-script login ABORT invalid TIMEOUT 15 name: billw word: wewpass ">"
"slip default"
interface async 10
dialer in-band
dialer map ip 10.55.0.1 modem-script dial system-script login 96837890

```

When a packet is received for 10.55.0.1, the first thing that happens is that the modem script is implemented. Table 1-2 shows the functions that are implemented with each expect-send pair in the modem script called “dial.”

**Table 1-2 Modem Script Execution**

Expect and Send Pairs	Execution
ABORT ERROR	End 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, send an “AT Z” command to the modem. Note the use of quotes to allow a space in the send string.
OK "ATDT \T"	Wait to see “OK.” Send “ATDT 96837890.”
TIMEOUT 30	Wait up to 30 seconds for next expect string.
CONNECT \c	Expect "connect," but don't send anything. Note that \c is effectively nothing. "" would have been nothing followed by a carriage return.

After the modem script is successfully executed, the login script is executed. Table 1-3 shows the functions that are executed with each expect-send pair in the system script “login.”

**Table 1-3 System Script Execution**

Expect and Send Pairs	Implementation
ABORT invalid	End the script execution if the message “invalid username or password” is displayed.
TIMEOUT 15	Wait up to 15 seconds.
name: billw	Look for “name:” and send “billw.” Using just “ame:” will help avoid any capitalization issues.
word: wewpass	Wait for “word:” and send the password.
">" "slip default"	Wait for the ts prompt, and then put the line into slip mode with its default address.

## Example of System Scripts and Modem Scripts

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 1.2.3.4, the communication server will dial the 91800 number using the usrobotics-v32 script, matching the regular expression in the modem chat script. Then the communication server will run the unix-slip chat script as the system script to log in.

If there is traffic for 4.3.2.1, the communication server will dial 8899 using usrobotics-v32, matching both the modem script and modem chat script regular expressions. The communication server 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 30
CONNECT \c
!
! Script for logging into a unix system and starting up slip:
chat-script unix-slip ABORT invalid TIMEOUT 15 name: billw word: wewpass ">"
"slip default"
    
```



```

! Script for logging into a cisco comm server and starting up TCP header
! compression
chat-script cisco-compressed...
!
Line 15
modem chat-script usrobotics-*
!
Interface async 15
dialer map ip 1.2.3.4 system-script unix-slip 918005551212
dialer map ip 4.3.2.1 modem-script *-v32 system-script cisco-compressed 8899

```

## Dial-on-Demand PPP Configuration Examples

The following example shows a configuration in which several aspects of DDR are used to provide DDR capabilities between local and remote communication servers. The following features are shown in this example.

- **dialer map** command
- PPP encapsulation
- CHAP
- Dialer rotary groups
- **pulse-time** command

See the “Interface Configuration Commands” chapter in the *Communication Server Command Reference* publication for a description of the **pulse-time** command.)

```

! Enable PPP encapsulation and CHAP on interface dialer 1.
interface dialer 1
ip address 131.108.2.1 255.255.255.0
ip address 131.126.4.1 255.255.255.0 secondary
encapsulation ppp
ppp authentication chap

! Specify dial-on-demand routing supported on a line and
! assign a set of access-list expressions.
dialer in-band
dialer group 1

! The first dialer map command indicates that calls between the remote site
! YYY and the central site will be placed at either end. The second dialer
! map command, with no dialer string, indicates that remote site ZZZ will call
! the central site but the central site will not call out.
dialer map ip 131.108.2.5 name YYY 1415553434
dialer map ip 131.126.4.5 name ZZZ

! The DTR pulse holds the DTR low for three seconds, so the modem can recognize
! that DTR has been dropped.
pulse-time 3

! Place asynchronous serial interfaces 1 and 2 in dialer rotary group 1. The
! interface subcommands applied to dialer rotary group 1 (for example,
! PPP encapsulation and CHAP) apply to these interfaces.
interface async 1
dialer rotary-group 1
interface async 2
dialer rotary-group 1

! CHAP passwords are specified for remote servers.
username YYY password theirsystem
username ZZZ password thatsystem

```

Figure 1-6 shows a configuration in which remote communication servers *YYY* and *ZZZ* and local communication server *XXX* are using dial-on-demand routing, as configured in the previous example. In this configuration, remote communication servers *YYY* and *ZZZ* can call communication server *XXX*. Communication server *XXX* has dialer string information only for communication server *YYY* and cannot call communication server *ZZZ*.

Figure 1-6 Dial-On Demand Routing Configuration

