Doc. No. 78-1467-02

Router Products Errata for IOS Release 10.1

December 5, 1994

This document provides corrections and additional information for the IOS Release 10 *Router Products Configuration Guide*, *Router Products Command Reference*, and *Enhanced IGRP Configuration Guide and Command Reference* publications.

The contents of this errata document are as follows:

Corrections to Chapter 3, System Images, Microcode Images, and Configuration Files

On page 3-24 of the configuration guide, add the following section before the "Configure a Router as a TFTP Server" section.

Manually Boot Using MOP

You can interactively boot system software using MOP. Typically, you would do this to verify that system software has been properly installed on the MOP boot server before configuring the router to automatically boot the system software image.

To manually boot the router using MOP, perform the following tasks:

Task		Command
Step 1	Restart the router from EXEC mode.	**reload**
Step 2	Press the Break key during the first 60 seconds while the system is starting up.	Break
Step 3	Manually boot the router using MOP.	**b mop** *filename* [*mac-address*] [*interface*]

Also, in the command reference, on page 3-4, change the syntax and syntax description of the **b** command to add the **b mop** version of the command:

To manually boot the router, use the **b** ROM monitor command:

b
b *filename* [*ip-address*]
b flash [*filename*]
b mop *filename* [*mac-address*] [*interface*]

Syntax Description

filename	Name of the system image from which to netboot.
ip-address	(Optional) IP address of the TFTP server on which the system image resides. If omitted, this value defaults to the IP broadcast address of 255.255.255.255.
flash [*filename*]	Boots the router from Flash memory with the optional filename of the image to load. The filename is case sensitive. If you omit a filename, the first valid file in Flash memory is loaded.
mop *filename*	Boots the router interactively using MOP. The filename is the name of the file image to load. Note that for VMS systems, the file on the host always ends with the .SYS extension; do not include this extension as part of the file name.
mac-address	(Optional) Hardware address of the host from which to load the image. If omitted, a broadcast message is sent to all MOP boot servers, and the first MOP server to indicate that it has the file becomes the server from which the router loads the image.

interface	(Optional) Interface from which the image is loaded. If omitted, a request is sent on all interfaces that have MOP enabled, and the interface that responds first is the one used to load the image.

On page 3-24 in the configuration guide, add the following section before the "Configure a Router as a TFTP Server" section.

Boot Systems That Have Dual-Bank Flash Memory

Some routers, such as the Cisco 4500, have two banks of Flash memory, referred to as dual-bank Flash memory. One bank of Flash memory contains the boot image, and the second bank contains the system image. The router uses the boot image to load router software from the network if configured to do so. The ROM monitor can start the system image directly. In the Cisco 4500, the system image is copied from Flash memory to RAM and runs from RAM.

Copy a Boot Image

You can retrieve a boot image from a TFTP server or from a MOP server. This image is copied into boot Flash memory. You can also copy the boot image from the boot Flash memory to a TFTP server.

To retrieve a boot image from a TFTP server, perform the following task in EXEC mode:

Task	Command
Copy a boot image from a TFTP server.	**copy tftp bootflash**

To retrieve a boot image from a MOP server, perform the following task in EXEC mode:

Task	Command
Copy a boot image from a MOP server.	**copy mop bootflash**

To copy a boot image from boot Flash memory to a TFTP server, perform the following task in EXEC mode:

Task	Command
Copy a boot image to a TFTP server.	**copy bootflash tftp**

Verify a Boot Image's Checksum

To verify the checksum of a boot image in Flash memory, perform the following task in EXEC mode:

Task	Command
Verify the checksum of a boot image.	**copy verify bootflash**

Erase Boot Flash Memory

To erase the contents of boot Flash memory, perform the following task at the EXEC prompt:

Task	Command
Erase boot Flash memory.	**copy erase bootflash**

Also, in the command reference, add the following new commands related to dual-bank Flash memory to Chapter 3:

copy bootflash tftp

To copy a boot image from Flash memory to a TFTP server, use the **copy bootflash tftp** EXEC command.

copy bootflash tftp

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

You can use this command only on routers that have two banks of Flash memory: one bank for the boot image and the second bank for the system image.

You might want to copy the boot image in order to save a backup copy of it or to verify that the copy in Flash memory is the same as on the original file.

Example

The following example illustrates how to use this command:

```
Router# copy bootflash tftp
Boot flash directory:
File  name/status
  1   c4500-xboot
[2557136 bytes used, 1637168 bytes available]

Address or name of remote host [255.255.255.255]? barney.cisco.com
Name of file to copy? c4500-xboot
Verifying checksum for 'c4500-xboot' (file # 1)...  [OK]
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!
```

Related Commands

copy erase bootflash
copy mop bootflash
copy tftp bootflash
copy verify bootflash
show bootflash

copy erase bootflash

To erase the boot image in Flash memory, use the **copy erase bootflash** EXEC command.

copy erase bootflash

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

You can use this command only on routers that have two banks of Flash memory: one bank for the boot image and the second bank for the system image.

Example

The following example erases the boot image in Flash memory:

```
copy erase bootflash
```

Related Commands

copy bootflash tftp
copy mop bootflash
copy tftp bootflash
copy verify bootflash
show bootflash

copy mop bootflash

To copy a boot image from a MOP server to Flash memory, use the **copy mop bootflash** EXEC command.

copy mop bootflash

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

You can use this command only on routers that have two banks of Flash memory: one bank for the boot image and the second bank for the system image.

The router prompts for the name of the image file. It provides an option to erase the existing boot image in Flash memory before writing the new image into Flash memory. If no free space is available, or if files have never been written to Flash memory, you must erase Flash memory before copying the MOP image.

You do not need to specify the address of a MOP server. The router automatically solicits a MOP boot server for the specified file by sending a multicast file-request message.

The copying process takes several minutes; the actual time differs from network to network.

Before booting from Flash memory, verify that the checksum of the image in Flash memory matches the checksum listed in the README file that was distributed with the boot software image. The checksum of the boot image in Flash memory is displayed when the **copy mop bootflash** command completes. The README file was copied to the MOP server automatically when you installed the boot software image.

Caution If the checksum values do not match, do not reboot the router. Instead, reissue the **copy mop bootflash** command and compress the checksums again. If the checksum is repeatedly wrong, copy the original boot software image back into Flash memory *before* you reboot the router from Flash memory.

Example

The following example shows how to use this command to copy the boot image *c4500-k*:

```
Router# copy mop bootflash
System bootflash directory:
File  name/status
  1   c4500-k
[4529048 bytes used, 3859560 bytes available]

Name of file to copy? c4500-k.101-beta
Copy c4500-ka from MOP server? [confirm] y
Erase flash device before writing? [confirm] y
Are you sure? [confirm] y
Erasing device... eeeeeeeeeevvvvvvvvv ... erased.

Loading c4500-k from MOP server: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Related Commands

copy bootflash tftp
copy erase bootflash
copy tftp bootflash
copy verify bootflash
show bootflash

copy tftp bootflash

To copy a boot image from a TFTP server to Flash memory, use the **copy tftp bootflash** EXEC command.

copy tftp bootflash

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

You can use this command only on routers that have two banks of Flash memory: one bank for the boot image and the second bank for the system image.

The router prompts for the address of the TFTP server and the name of the file. It provides an option to erase the existing boot image in Flash memory before writing the new image into Flash memory. The copying process takes several minutes; the actual time differs from network to network.

Before booting from Flash memory, verify that the checksum of the image in Flash memory matches the checksum listed in the README file that was distributed with the boot software image. The checksum of the boot image in Flash memory is displayed when the **copy tftp bootflash** command completes. The README file was copied to the TFTP server automatically when you installed the boot software image.

Caution If the checksum values do not match, do not reboot the router. Instead, reissue the **copy tftp bootflash** command and compare the checksums again. If the checksum is repeatedly wrong, copy the original boot software image back into Flash memory *before* you reboot the router from Flash memory.

Example

The following example shows how to use this command:

```
Router# copy tftp bootflash
Boot flash directory:
File  name/status
  1   old-c4500-xboot
[2557136 bytes used, 1637168 bytes available]

Address or name of remote host [255.255.255.255]? barney.cisco.com
Name of file to copy? c4500-xboot
Copy c4500-xboot from BARNEY.CISCO.COM? [confirm] y
Checking for file 'c4500-xboot' on BARNEY.CISCO.COM... [OK]

Erase flash device before writing? [confirm] y
Are you sure? [confirm] y
Erasing device... eeeeeeeeeevvvvvvvvv ... erased.

Loading c4500-xboot from 198.92.30.32: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!!!!!!!!!!!!

[OK - 1387269/4194304 bytes]

Verifying checksum... (0x142A) [OK]
```

Related Commands

copy bootflash tftp
copy erase bootflash
copy mop bootflash
copy verify bootflash
show bootflash

copy verify bootflash

To verify the checksum of a boot image in Flash memory, use the **copy verify bootflash** EXEC command.

copy verify bootflash

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

You can use this command only on routers that have two banks of Flash memory: one bank for the boot image and the second bank for the system image.

Each boot software image that is distributed on disk uses a single checksum for the entire image. This checksum is displayed only when the image is copied into Flash memory; it is not displayed when the image file is copied from one disk to another.

The README file, which is included with the image on the disk, lists the name, file size, and checksum of the image. Review the contents of the README file before loading or duplicating the new image so that you can verify the checksum when you copy it into Flash memory or onto a server.

To display the contents of Flash memory, use the **show flash** command. The Flash contents listing does not include the checksum of individual files. To recompute and verify the image checksum after the image has been copied into Flash memory, use the **copy verify bootflash** command. When you enter the command, the system prompts you for the filename to verify. By default, it prompts for the last file (most recent) in Flash memory. Press Return to recompute the default file checksum, or enter the name of a different file at the prompt.

Example

The following example illustrates how to use this command:

```
Router# copy verify bootflash
Name of file to verify [c4500-k]?
Boot flash directory:
```

```
File  name/status
  1   c4500-xboot
[1387336 bytes used, 2806968 bytes available]

Verifying checksum for 'c4500-xboot' (file # 1)...  [OK]
```

Related Commands

copy bootflash tftp
copy erase bootflash
copy mop bootflash
copy tftp bootflash
show bootflash

show bootflash

To verify boot Flash memory, use the **show bootflash** EXEC command.

show bootflash

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

You can use this command only on routers that have two banks of Flash memory: one bank for the boot image and the second bank for the system image.

The **show bootflash** command displays the type of boot Flash memory present, any files that might currently exist in boot Flash memory, and the amount of boot Flash memory used and remaining.

Sample Display

The following is sample output from the **show bootflash** command:

```
Router# show bootflash
Boot flash directory:
File  name/status
  1   c4500-xboot
[1387336 bytes used, 2806968 bytes available]
```

Table 1 describes the fields shown in the output.

Table 1 Show Bootflash Field Descriptions

Field	Description
Boot File	Number of the boot file.
flash directory: name/status	Name and status of the boot file. The status is displayed if appropriate and can be one of the following: • [deleted]—File has been deleted. • [invalid checksum]—File has an incorrect checksum.

On page 3-30 of the configuration guide, add the following section before the "Verify the Image in Flash Memory" section.

Copy System Images to Flash Memory Using MOP

To use MOP to copy a system image to Flash memory, perform the following task at the EXEC prompt:

Task	Command
Copy a boot image using MOP.	**copy mop flash**

Also, in the command reference, add the description of the **copy mop flash** command after page 3-19.

copy mop flash

To use MOP to copy a system image to Flash memory, use the **copy mop flash** EXEC command.

copy mop flash

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

MOP must be enabled on the relevant interfaces before you can use this command.

Examples

The following example illustrates how to use the **copy mop flash** command:

```
Router# copy mop flash
System flash directory:
File  name/status
1  old-c4500-k
[2264000/4194304 bytes free/total]
```

```
Name of file to copy ? c4500-k
Copy c4500-k from MOP server into flash memory ?  [confirm] y
Erase flash device before writing? [confirm] n

Loading c4500-k from MOP server: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!
```

Note Make sure you have enough Flash memory space before entering **n** at the "Erase flash device before writing?" prompt. The Flash directory information displayed after you enter the **copy mop flash** command and before the prompts indicates the amount of free and total space in Flash memory.

On page 3-30 of the configuration guide, add the following section after the section "Verify the Image in Flash Memory."

Corrections to Chapter 5, System Management

In the *Router Products Configuration Guide*, on pages 5-10 and 5-11, replace the sections "Set the Cisco 7000 Calendar" and "Monitor Time Services" with these revised sections. The Cisco 4500 also includes system calendar hardware.

Set the System Calendar Capabilities

In addition to a system clock, the Cisco 4500 and Cisco 7000 hardware provides a system calendar that can set the system time and control the system clock, as well as enable the router to act as a time service for the network.

You can complete the following tasks to enable the Cisco 4500 and Cisco 7000 calendar capabilities:

- Set the System Calendar
- Set the Router as a Network Time Source
- Set the System Clock from the Router's Calendar
- Set the Router's Calendar from the System Clock

Set the System Calendar

The router calendar maintains time separately from the system clock. It continues to run when the system is restarted or power is turned off. Typically, it will only need to be manually set once, when the system is first installed. If time is available from an external source using NTP, the calendar can be updated from the system clock instead.

If you do not have an external time source, perform the following task in EXEC mode to set the system calendar:

Task	Command
Set the router calendar.	**calendar set** *hh:mm:ss day month year* or **calendar set** *hh:mm:ss month day year*

Set the Router as a Network Time Source

Although the system clock is always initialized from the router calendar when the system is restarted, by default it is not considered to be authoritative and so will not be redistributed with NTP or VINES Time Service. To make the router's calendar be authoritative, complete the following task in global configuration mode:

Task	Command
Enable the router to act as a valid time source to which network peers can synchronize.	**clock calendar-valid**

For an example of making the router's calendar authoritative, see the section "Clock, Calendar, and NTP Configuration Example" at the end of this chapter.

Set the System Clock from the Router's Calendar

To set the system clock to the new calendar setting, perform the following task in EXEC mode:

Task	Command
Set the system clock from the calendar.	**clock read-calendar**

Set the Router's Calendar from the System Clock

To update the calendar with the new clock setting, perform the following task in EXEC mode:

Task	Command
Set the calendar from the system clock.	**clock update-calendar**

Monitor Time Services

To monitor clock, calendar, and NTP EXEC services, perform one or more of the following tasks in EXEC mode:

Task	Command
Display the current calendar time (for the Cisco 4500 and Cisco 7000 only).	**show calendar**
Display the current system clock time.	**show clock** [**detail**]
Show the status of NTP associations.	**show ntp associations [detail]**
Show the status of NTP.	**show ntp status**

In the *command reference*, note that you can use the **calendar set** (page 5-6), **clock calendar-valid** (page 5-7), **clock read-calendar** (page 5-8), **clock update-calendar** (page 5-13), and **show calendar** (page 5-75) on Cisco 4500 routers as well as on Cisco 7000 series routers.

Corrections to Chapter 6, Interfaces

In the configuration guide, on page 6-6, replace the section "Fiber Distributed Data Interface (FDDI)" with this revised section, which adds mention of support for SMT Version 7.3.

The Fiber Distributed Data Interface (FDDI) is an ANSI-defined standard for timed 100-Mbps token passing over fiber-optic cable. An FDDI network consists of two counter token-passing fiber-optic rings. On most networks, the primary ring is used for data communication and the secondary ring is used as a hot standby. The FDDI standard sets total fiber lengths of 2 kilometers for multimode fiber and 10 kilometers for single-mode fiber, both of which are supported by our FDDI interface controller. (The maximum circumference of the FDDI network is only half the specified kilometers because of the *wrapping* or looping back of the signal that occurs during fault isolation.)

The FDDI standard allows a maximum of 500 stations with a maximum distance between active stations of two kilometers. The FDDI frame can contain a minimum of 22 bytes and a maximum of 4500 bytes. Our implementation of FDDI supports Station Management (SMT) Version 7.3 of the X3T9.5 FDDI specification, offering a single MAC dual-attach interface that supports the fault-recovery methods of the dual attachment stations (DASs). The midrange platforms also support single-attach stations (SASs).

We also provide support for some of the FDDI MIB variables as described in RFC 1285, "FDDI Management Information Base," published in January 1992 by Jeffrey D. Case of the University of Tennessee and SNMP Research, Inc. One such variable that we support is *snmpFddiSMTCFState*.

In the command reference on page 6-24, add the following paragraph to the "Usage Guidelines" section for the **compress predictor** command:

When using compression, you should adjust the MTU for the serial interface and the LAPB N1 parameter as shown in the example to avoid informational diagnostics regarding excessive MTU or N1 sizes.

Replace the example in the "Example" section with the following:

```
interface serial 0
encapsulation lapb
compress predictor
mtu 1509
lapb n1 12072
```

In Chapter 6 of the command reference, add the following new commands.

carrier-delay

To set the carrier delay on a serial interface, use the **carrier-delay** interface configuration command. To return to the default carrier delay value, use the **no** form of this command.

carrier-delay *seconds*
no carrier-delay [*seconds*]

Syntax Description

seconds	Time, in seconds, for the system to change states. This can be an integer in the range 0 to 60. The default is 2 seconds. When choosing a value, we recommend that you choose a lower one rather than a higher one.

Default

2 seconds

Command Mode

Interface configuration

Example

The following example changes the carrier delay to 5 seconds:

```
interface serial 0
carrier-delay 5
```

serial restart-time

To set the restart timer on a serial interface, use the **serial restart-time** interface configuration command. To return to the default restart timer value, use the **no** form of this command.

serial restart-time *seconds*
no serial restart-time [*seconds*]

Syntax Description

seconds	Time, in seconds, to wait for a serial interface to come up before resetting the interface. This can be an integer in the range 0 to 900. The default is 15 seconds. A value of 0 means that the serial interface is never reset.

Default

15 seconds

Command Mode

Interface configuration

Usage Guidelines

On serial interfaces where resetting the interface could cause an incoming call to be dropped, you should set the restart delay to 0.

You cannot use this command on channelized T1 and E1 interfaces.

Example

The following example eliminates the restart delay so that the serial interface is never reset:

```
interface serial 0
serial restart-time 0
```

Corrections to Chapter 8, DDR

In the command reference on page 8-15, for the **dialer-list list** command, delete the final line from the supported access list types and numbers table, which indicates Novell SAP and the access-list number range 1000-1099.

In the command reference on page 8-23, revise the syntax description of the *hostname* argument for the **dialer map name** command as follows:

hostname	Name or ID of the remote device (usually the host name).

Change the "Default" section to read as follows:

No dialer map is configured.

Add the following sentence to the "Usage Guidelines" section:

For routers with ISDN interfaces, if calling line identification (CLI/ANI/caller id) is provided, the *hostname* field may contain the number that calling line ID provides.

In the "Related Commands" section, delete the **dialer map modem** command.

Corrections to Chapter 9, Frame Relay

In the command reference, add the **frame-relay broadcast-queue** command after the **encapsulation frame-relay** command on page 9-3, as follows:

frame-relay broadcast-queue

To create a special queue for a specified interface to hold broadcast traffic that has been replicated for transmission on multiple DLCIs, use the **frame-relay broadcast-queue** interface configuration command.

frame-relay broadcast-queue *size byte-rate packet-rate*

Syntax Description

size	Number of packets to be held in the broadcast queue. The default is 64 packets.
byte-rate	Maximum number of bytes to be transmitted per second. The default is 256000 bytes per second.
packet-rate	Maximum number of packets to be transmitted per second. The default is 36 packets per second.

Default

The default values are as follows:

size—64 bytes
byte-rate—256000 bytes per second
packet-rate—36 packets per second

Command Mode

Interface configuration

Usage Guidelines

For purposes of the Frame Relay broadcast queue, broadcast traffic is defined as packets that have been replicated for transmission on multiple DLCIs; however, broadcast traffic does not include the original routing packet or SAP packet, which passes through the normal queue. Due to timing sensitivity, bridged broadcasts and spanning tree packets are sent through the normal queue.

The Frame Relay broadcast queue is managed independently of the normal interface queue. It has its own buffers and a configurable service rate.

A broadcast queue is given a maximum transmission rate (throughput) limit measured in bytes per second and packets per second. The queue is serviced to ensure that only this maximum is provided. The broadcast queue has priority when transmitting at a rate below the configured maximum, and hence has a guaranteed minimum bandwidth allocation. The two transmission rate limits are intended to avoid flooding the interface with broadcasts. The actual limit in any second is the first rate limit that is reached.

Given the transmission rate restriction, additional buffering will be required to store broadcast packets. The broadcast queue is configurable to store large numbers of broadcast packets.

You should set the queue size to avoid loss of broadcast routing update packets. The exact size will depend on the protocol being used and the number of packets required for each update. To be safe, set the queue size so that one complete routing update from each protocol and for each DLCI can be stored. Consider starting with 20 packets per DLCI.

In general, the byte rate should be less than both of the following:

- *N*/4 times the minimum remote access rate (measured in *bytes* per second), where N is the number of DLCIs to which the broadcast must be replicated
- 1/4 the local access rate (measured in *bytes* per second)

The packet rate is not critical if you set the byte rate conservatively. As a general rule, set the packet rate assuming 250-byte packets.

Example

The following example specifies a broadcast queue to hold 80 packets, to have a maximum byte transmission rate of 240,000 bytes per second, and to have a maximum packet transmission rate of 160 packets per second:

```
frame-relay broadcast-queue 80 240000 160
```

On page 9-6 of the command reference, revise the “Default” section of the **frame-relay inverse-arp** command to read “Enabled.”

On page 9-7 of the configuration guide, revise the section "Set the LMI Type" as follows:

Set the LMI Type

You can set one of three types of LMIs on our router: ANSI T1.617 Annex D, Cisco, and ITU-T Q.933 Annex A. To do so, perform the following task in interface configuration mode:

Task	Command
Set the LMI type.	**frame-relay lmi-type** {**ansi** | **cisco** | **q933a**}

For an example of how to set the LMI type, see the section "Example of Configuring a Pure Frame Relay DCE" later in this chapter.

On page 9-8 of the configuration guide, revise the section "Select Frame Relay Inverse ARP" as follows:

Select Frame Relay Inverse ARP

Frame Relay Inverse ARP is a method of building dynamic routes in Frame Relay networks running AppleTalk, Banyan VINES, DECnet, IP, Novell IPX, and XNS. Inverse ARP allows the router to discover the protocol address of a device associated with the virtual circuit. Inverse ARP is used instead of the **frame-relay map** command, which allows you to define the mappings between a specific protocol and address and a specific DLCI (see the section "Establish Mapping" earlier in this chapter for more information).

Inverse ARP is enabled by default. Configure Inverse ARP if you want to configure an interface for multipoint communication that was previously configured for point-to-point. You would not need to select Inverse ARP if you have a point-to-point interface, because there is only a single destination and discovery is not required.

To select Inverse ARP, perform the following task in interface configuration mode:

Task	Command
Select Frame Relay Inverse ARP.	**frame-relay inverse-arp** *protocol dlci*

On page 9-9 of the configuration guide, add the following section after the section "Associate a DLCI with a Subinterface."

Create a Broadcast Queue for an Interface

Very large Frame Relay networks might have performance problems when many DLCIs terminate in a single router and the router must replicate routing updates and service advertising updates on each DLCI. The updates can consume access-link bandwidth and cause significant latency variations in user traffic; the updates can also consume interface buffers and lead to higher packet rate loss for both user data and routing updates.

To avoid such problems, you can create a special broadcast queue for an interface. The broadcast queue is managed independently of the normal interface queue, has its own buffers, and has a configurable size and service rate.

A broadcast queue is given a maximum transmission rate (throughput) limit measured in both bytes per second and packets per second. The queue is serviced to ensure that no more than this maximum is provided. The broadcast queue has priority when transmitting at a rate below the configured maximum, and hence has a guaranteed minimum bandwidth allocation. The two transmission rate limits are intended to avoid flooding the interface with broadcasts. The actual transmission rate limit in any second is the first of the two rate limits that is reached.

To create a broadcast queue, complete the following task in interface configuration mode:

Task	Command
Create a broadcast queue for an interface.	**frame-relay broadcast-queue** *size byte-rate packet-rate*

On page 9-14 of the command reference, for the **frame-relay lmi-type** command, revise the command syntax and syntax description as follows:

frame-relay lmi-type {**ansi** | **cisco** | **q933a**}
no frame-relay lmi-type {**ansi** | **q933a**}

Syntax Description

ansi	Annex D defined by ANSI standard T1.617
cisco	Group of 4 LMI
q933a	ITU-T[1] Q.933 Annex A

1. The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) carries out the functions of the former Consultative Committee for International Telegraph and Telephone (CCITT).

On page 9-16 and 9-17 of the configuration guide, revise the configurations for Router A and Router C to indicate **frame-relay lmi-type q933a**. The complete examples are as follows:

Configuration for Router A

```
frame-relay switching
!
interface Ethernet0
no ip address
shutdown
!
interface Ethernet1
no ip address
shutdown
!
interface Ethernet2
no ip address
shutdown
!
interface Ethernet3
no ip address
shutdown
!
interface Serial0
ip address 131.108.178.48 255.255.255.0
shutdown
```

```
!
interface Serial1
no ip address
encapsulation frame-relay
frame-relay intf-type dce
frame-relay lmi-type ansi
frame-relay route 100 interface serial 2 200
!
interface Serial2
no ip address
encapsulation frame-relay
frame-relay intf-type nni
frame-relay lmi-type q933a
frame-relay route 200 interface serial 1 100
clockrate 2048000
!
interface Serial3
no ip address
shutdown
```

Configuration for Router C

```
frame-relay switching
!
interface Ethernet0
no ip address
shutdown
!
interface Ethernet1
no ip address
shutdown
!
interface Ethernet2
no ip address
shutdown
!
interface Ethernet3
no ip address
shutdown
!
interface Serial0
ip address 131.108.187.84 255.255.255.0
shutdown
!
interface Serial1
no ip address
encapsulation frame-relay
frame-relay intf-type dce
frame-relay route 300 interface serial 2 200
!
interface Serial2
no ip address
encapsulation frame-relay
frame-relay intf-type nni
frame-relay lmi-type q933a
frame-relay route 200 interface serial 1 300
!
interface Serial3
no ip address
shutdown
```

In the command reference, for the **show frame-relay map** command on page 9-25, in the "Show Frame-Relay Map Field Descriptions" table, revise the description of the CISCO field as follows:

Field	Description
CISCO	Specifies the encapsulation type: CISCO or IETF.

Corrections to Chapter 11, X.25 and LAPB

In the configuraiton guide, on page 11-7, in Table 11-1, revise the range for **lapb n1** *bits* to 1088–32840 bits.

In the command reference, on page 11-17, in the "Syntax Description," revise the description for **lapb n1** *bits* to the following:

bits	Number of bits. It can be a number from 1088 through 32840. It must be a multiple of eight.

Corrections to Chapter 14, Banyan VINES

On page 14-3 in the configuration guide, replace the section "Enable VINES Routing on the Router" with the following:

Enable VINES Routing on the Router

To enable VINES routing on the router, perform the following task in global configuration mode:

Task	Command
Enable VINES RTP routing on the router.	**vines routing** [*address*]

Enabling VINES routing on the router starts the VINES Routing Update Protocol (RTP) by default.

To enable the Sequenced Routing Update Protocol (SRTP) also, perform the following task in global configuration mode:

Task	Command
Enable VINES SRTP routing on the router.	**vines srtp-enabled**

For an example of how to enable VINES routing, see the section "Typical VINES Network Configuration Example" later in this chapter.

After page 14-61 of the command reference, add the following description of the **vines srtp-enabled** command:

vines srtp-enabled

To enable the Sequenced Routing Update Protocol (SRTP), use the **vines srtp-enabled** global configuration command. To disable SRTP, use the **no** form of this command.

vines srtp-enabled
no vines srtp-enabled

Syntax Description

This command has no arguments or keywords.

Default

By default, the router runs Banyan's Routing Update Protocol (RTP) routing protocol only.

Command Mode

Global configuration

Usage Guidelines

When SRTP is enabled, the router dynamically determines whether it needs to send RTP messages, SRTP messages, or both.

Example

The following example enables SRTP on serial interface 0:

```
interface serial 0
vines routing
vines srtp-enabled
```

Related Command

vines routing

Corrections to Chapter 16, IP

On page 16-111 of the command reference, for the **show ip route** command, add the asterisk (*) and its description to the end of the "Show IP Route Field Descriptions" table, as follows:

Field	Description
*	Indicates the last path used when a packet was forwarded. It pertains only to the nonfast-switched packets. However, it does not indicate what path will be used next when forwarding a nonfast-switched packet except when the paths are equal cost.

Corrections to Chapter 17, IP Routing Protocols

On page 17-9 of the command reference and on page 17-12 of the configuration guide, revise the syntax of the **area virtual-link** command as follows:

area *area-id* **virtual-link** *router-id* [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**transmit-delay** *seconds*] [**dead-interval** *seconds*] [**authentication-key** *password*]
no area *area-id* **virtual-link** *router-id* [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**transmit-delay** *seconds*] [**dead-interval** *seconds*] [**authentication-key** *password*]

On page 17-10 of the command reference, add the following sentence to the Syntax Description of the *password* argument for the **area virtual-link** command:

The 8 bytes of password are encrypted in the configuration file if the **service password-encryption** command is enabled.

On page 17-10 of the command reference, add the following sentence to the Usage Guidelines of the **area virtual-link** command:

Any argument specified after **authentication-key** *password* is ignored. Therefore, specify any optional arguments before **authentication-key**.

On page 17-27 of the command reference, add the following sentence to the *weight* argument of the **distance** command:

A distance of 255 is the maximum possible distance, and any route with that distance will not be installed in the routing table.

On page 17-29 of the command reference, add the following sentence to the *external-distance*, *internal-distance*, and *local-distance* arguments of the **distance bgp** command:

A distance of 255 is the maximum possible distance, and any route with that distance will not be installed in the routing table.

On page 17-44 of the command reference, for the **ip ospf cost** command, change the description of the *cost* argument to the following:

Unsigned integer value expressed as the link state metric. The range is from 1 to 65535.

On page 17-110 of the command reference, change the syntax of the **redistribute** command to the following:

redistribute *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*]
[**metric-type** *type-value*] [**match** {**internal** | **external1** | **external2**}]
[**tag** *tag-value*] [**route-map** *map-tag*] [**weight** *weight*] [**subnets**]
no redistribute *protocol* **[***process-id*] **{level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*]
[**metric-type** *type-value*] [**match** {**internal** | **external1** | **external2**}]
[**tag** *tag-value*] [**route-map** *map-tag*] [**weight** *weight*] [**subnets**]

On page 17-111 of the command reference, change the description of the keywords **match internal | external** and **external** *type-value* to the following:

match {**internal** | **external1** | **external2**}	(Optional) For OPSF, the criteria by which OSPF routes are redistributed into other routing domains. Can be one of the following: **internal**—Routes that are internal to a specific autonomous system. **external1**—Routes that are external to the autonomous system, but are imported into OSPF as type 1 external routes. **external2**—Routes that are external to the autonomous system, but are imported into OSPF as type 2 external routes.

On page 17-112 of the command reference, in the "Usage Guidelines" section of the **redistribute** command, add the following paragraph:

When routes are redistributed into OSPF and no metric is specified in the **metric** keyword, the default metric that OSPF uses is 20 for routes from all protocols except BGP route, which gets a metric of 1.

Corrections to Chapter 19, Novell IPX

On page 19-4 of the command reference and on page 19-6 of the configuration guide, the syntax of the extended **access-list** command is wrong. Change it to the following:

access-list *access-list-number* {**deny** | **permit**} *protocol* [*source-network*][[[**.***source-node*] *source-node-mask*] | [.*source-node source-network-mask.source-node-mask*]] [*source-socket*] [*destination.network*]**[**[[**.***destination-node*] *destination-node-mask*] | [**.***destination-node destination-network-mask.destination-nodemask*]] [*destination-socket*]

On page 19-21 of the command reference, for the **ipx delay** command, change the "Default" section to the following:

Determined from the delay configured on the interface with the **delay** command. It is (the interface delay + 333) / 334. Therefore, unless you change the delay by a value greater than 334, you will not notice a difference.

On page 19-21 of the command reference, add the following sentence to the "Usage Guidelines" section:

If the link is an IPXWAN link, it determines its delay dynamically and the **ipx delay** command has no effect.

On page 19-30 of the command reference, for the **ipx ipxwan** command, add the following to the *retry-attempts* argument in the "Syntax Description" section:

The router intentionally ignores the IPXWAN retry counter. The router continues to send out TIMER_REQUEST packets until it receives a TIMER_RESPONSE packet.

On page 19-37 of the command reference, for the **ipx network** command, revise the example as follows, because each secondary must have a different encapsulation specified:

Example

The following example configures an interface that has four logical networks:

```
interface ethernet 0
ipx network 0123
ipx encapsulation snap
ipx network 0234 encapsulation sap secondary
ipx network 0345 encapsulation arpa secondary
ipx network 0456 encapsulation novell-ether secondary
```

On page 19-74 of the command reference, for the **show ipx route** command, add the following code and description to Table 19-9, "Show IPX Route Field Descriptions":

Field	Description
W	Directly connected route determined via IPXWAN.

Corrections to Chapter 23, STUN

On pages 23-10 and 23-11 of the command reference, remove the **stun cos-enable** command.

On page 23-13 of the configuration guide, remove the section "Enable Class of Service," which documented the **stun cos-enable** command.

Corrections to *Enhanced IGRP Configuration Guide and Command Reference*

On page 3-12, in the section "RIP and IP Enhanced IGRP Redistribution Example," revise "Example 2: Complex Redistribution" as follows:

Example 2: Complex Redistribution

The most complex redistribution case is one in which *mutual* redistribution is required between an IGP (in this case IP Enhanced IGRP) and BGP.

Suppose that BGP is running on a router somewhere else in AS 1, and that the BGP routes are injected into IP enhanced IGRP routing process 1. You must use filters to ensure that the proper routes are advertised. The example configuration for router R1 illustrates use of access filters and a distribution list to filter routes advertised to BGP neighbors. This example also illustrates configuration commands for redistribution between BGP and IP enhanced IGRP.

```
! Configuration for router R1:
router bgp 1
network 131.108.0.0
neighbor 192.5.10.1 remote-as 2
neighbor 192.5.10.15 remote-as 1
neighbor 192.5.10.24 remote-as 3
redistribute eigrp 1
distribute-list 1 out eigrp 1
!
! All networks that should be advertised from R1 are controlled with access lists:
!
access-list 1 permit 131.108.0.0
access-list 1 permit 150.136.0.0
```

```
access-list 1 permit 128.125.0.0
!
router eigrp 1
network 131.108.0.0
network 192.5.10.0
redistribute bgp 1
```

On page 3-12, in the section "Default Metric Values Redistribution Example," delete the sentence "Figure 3-2 illustrates this type of redistribution." On page 3-13, delete Figure 3-2, "Assigning Metrics for Redistribution."

On page 4-4, in the section "Control SAP Updates," add the following task to the first task table:

Task	Command
Send SAP updates only when a change in the SAP table occurs, and send SAP changes only.	**ipx sap-incremental eigrp** *autonomous-system-number* **rsup-only**

On page 7-9, change the command syntax to the following:

ipx sap-incremental eigrp *autonomous-system-number* [**rsup-only**]
no ipx sap-incremental eigrp *autonomous-system-number* [**rsup-only**]

Add the following keyword to the "Syntax Description" section:

rsup-only (Optional) Indicates that the system uses enhanced IGRP on this interface to carry reliable SAP update information only.

Add the following paragraph to the "Usage Guidelines" section:

To reduce SAP traffic by sending partial SAP updates, specify the **rsup-only** keyword. SAP updates are then sent only when changes occur, and only changes are sent. This feature works with existing IPX RIP networks and IPX enhanced IGRP networks.

On page 7-13, replace the existing sample display with the following sample display:

```
Router# show ipx eigrp neighbors
IPX EIGRP Neighbors for process 200
H  Address                 Interface     Hold   Uptime   Q    Seq    SRTT  RTO
                                         (secs) (h:m:s)  Cnt  Num    (ms)  (ms)
6  90.0000.0c02.096e       Tunnel44444   13     0:30:57  0    21     9     20
5  80.0000.0c02.34f2       Fddi0         12     0:31:17  0    62     14    28
4  83.5500.2000.a83c       TokenRing2    13     0:32:36  0    626    16    32
3  98.0000.3040.a6b0       TokenRing1    12     0:32:37  0    43     9     20
2  80.0000.0c08.cbf9       Fddi0         12     0:32:37  0    624    19    38
1  85.aa00.0400.153c       Ethernet2     12     0:32:37  0    627    15    30
0  82.0000.0c03.4d4b       Hssi0         12     0:32:38  0    629    12    24
```

On page 7-14, in Table 7-1, add the following field and description after process 200:

Field	Description
H	Handle. An arbitrary and unique number inside therouter that identifies the neighbor.

This document is to be used in conjunction with the IOS Release 10 *Router Products Configuration Guide*, *Router Products Command Reference*, and *Enhanced IGRP Configuration Guide and Command Reference* publications.