de-bit through exp

de-bit

To set Frame Relay discard-eligible (DE) bit mapping for FRF.5 and FRF.8 network interworking, use the **de-bit command in FRF.5 connect configuration mode or FRF.8 connect configuration mode**. To disable or reset Frame Relay DE bit mapping, use the **no** form of this command.

de-bit {**0**| **1**| **map-clp**}

no de-bit {**0**| **1**| **map-clp**}

Syntax Description

0	Sets the DE field in the Frame Relay header to 0. This keyword may be used only for FRF.8.
1	Sets the DE field in the Frame Relay header to 1. This keyword may be used only for FRF.8.
map-clp	DE field in the Frame Relay header is set to 1 if one or more cells that belong to a frame have their cell loss priority (CLP) field set. This is the default setting. This keyword may be used for FRF.5 or FRF.8. **Note** **The map-clp** keyword is the only one available for FRF.5.

Command Default

map-clp

Command Modes

FRF.5 connect configuration FRF.8 connect configuration

Command History

Release	Modification
12.1(2)T	This command was introduced.
12.2(8)YN	Enhanced QoS features were added for Cisco 1720, Cisco 1750, Cisco 1751, Cisco 1760, Cisco 2610XM-2651XM, Cisco 3640, Cisco 3640A, and Cisco 3660.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.3(2)T	This command was integrated into Cisco IOS Release 12.3(2)T for the following platforms: Cisco 1721, Cisco 2610-2651, Cisco 2610XM-2651XM, Cisco 2691, Cisco 3620, and Cisco 3660.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Release	Modification
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

In the default state, the DE bit in the Frame Relay header is set to 1 when one or more ATM cells that belong to a frame have their cell loss priority (CLP) field set to 1 or when the DE field of the Frame Relay service-specific convergence sublayer (FR-SSCS) protocol data unit (PDU) is set to 1.

When the **no de-bit**command and **map-clp** keyword are entered, the FR-SSCS PDU DE field is copied unchanged to the Q.922 core frame DE field, independently of CLP indications received at the ATM layer.

Examples

The following example creates a connection between the virtual circuit (VC) group named “friends” and ATM PVC 0/32 and configures FR DE field mapping to match the ATM CLP field:

```
Router(config)#
vc-group friends
Router(config-vc-group)#
serial1/0 16 16
Router(config-vc-group)#
serial1/0 17 17
Router(config-vc-group)#
serial1/0 18 18
Router(config-vc-group)#
serial1/0 19 19
Router(config)#
interface atm3/0
R
outer(config-if)# pvc 0/32
R
outer(config-if-atm-vc)# encapsulation aal5mux frame-relay
Router (config-if-atm-vc)# exit
Router (config-if)# exit
Router(config)#
connect vc-group friends atm3/0 0/32
R
outer(config-frf5)# de-bit map-clp
```

Related Commands

Command	Description
clp-bit	Sets the ATM CLP field in the ATM cell header.
connect (FRF.5)	Configures an FRF.5 one-to-one connection or one-to-many connection between two Frame Relay end users over an intermediate ATM network.
connect (FRF.8)	Configures an FRF.8 one-to-one mapping between a Frame Relay DLCI and an ATM PVC.
vc-group	Assigns multiple Frame Relay DLCIs to a VC group.

de-bit map-clp

To set Frame Relay discard eligible (DE) bit mapping for FRF.5 network interworking, use the **de-bit map-clp** command in FRF.5 connect mode. To disable or reset Frame Relay DE bit mapping, use the **no** form of this command.

de-bit map-clp

no de-bit map-clp

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

FRF.5 connect configuration

Command History

Release	Modification
12.1(2)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

In the default state, the DE bit in the Frame Relay header is set to 1 when one or more ATM cells belonging to a frame have their cell loss priority (CLP) field set to 1, or when the DE field of the Frame Relay service specific convergence sublayer (FR-SSCS) protocol data unit (PDU) is set to 1.

When the **no de-bit map-clp** command is entered, the FR-SSCS PDU DE field is copied unchanged to the Q.922 core frame DE field, independent of CLP indications received at the ATM layer.

Examples

The following example creates a connection that connects the virtual circuit (VC) group named friends to ATM PVC 0/32 and configures FR DE field mapping to match the ATM CLP field:

```
Router(config)#
vc-group friends
Router(config-vc-group)#
serial0 16 16
Router(config-vc-group)#
serial0 17 17
Router(config-vc-group)#
serial0 18 18
Router(config-vc-group)#
serial0 19 19
```

```
Router(config)#
interface atm3/0
Router
(config-if)# pvc 0/32
Router
(config-if-atm-vc)# encapsulation aal5mux frame-relay
Router(config)#
connect vc-group friends atm3/0 0/32
Router
(config-frf5)# de-bit map-clp
```

Related Commands

Command	Description
clp-bit	Sets the ATM CLP field in the ATM cell header.
connect (FRF.5)	Connects a Frame Relay DLCI or VC group to an ATM PVC.
vc-group	Assigns multiple Frame Relay DLCIs to a VC group.

debug l4f

To enable troubleshooting for Layer 4 Forwarding (L4F) flows, use the **debug l4f** command in privileged EXEC mode. To disable the troubleshooting, use the **no** form of this command.

debug l4f {**api**| **flow-db**| **flows**| **packet** {**all**| **detail**| **injection**| **interception**| **proxying**| **spoofing**}| **test-app**| **trace-db-api**| **trace-db-flow**| **trace-engine**}

no debug l4f {**api**| **flow-db**| **flows**| **packet** {**all**| **detail**| **injection**| **interception**| **proxying**| **spoofing**}| **test-app**| **trace-db-api**| **trace-db-flow**| **trace-engine**}

Syntax Description

api	Toggles L4F API debugging.
flow-db	Toggles L4F flow database debugging.
flows	Toggles L4F flows debugging.
packet	Toggles L4F packet debugging.
all	Toggles all L4F packet debugging.
detail	Toggles L4F packet detail debugging.
injection	Toggles L4F packet injection debugging.
interception	Toggles L4F packet interception debugging.
proxying	Toggles L4F packet proxying debugging.
spoofing	Toggles L4F packet spoofing debugging.
test-app	Toggles L4F test application debugging.
trace-db-api	Toggles L4F database API debugging.
trace-db-flow	Toggles L4F database flow debugging.
trace-engine	Toggles L4F API tracing debugging.

Command Default

L4F debugging is off.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(2)T	This command was introduced.

Usage Guidelines

Use this command to enable debugging for Layer 4 forwarding flows.

Examples

The following example shows how to enable debugging for L4F packets:

```
Router# debug l4f packet all
```

Related Commands

Command	Description
show l4f	Displays the flow database for L4F.

debug platform hardware qfp active interface frame-relay multilink

To debug the multilink frame-relay interfaces in the Cisco QuantumFlow Processor (QFP), use the **debug platform hardware qfp interface frame-relay mulitlink**command in the Privileged EXEC mode. To disable this form of debugging, use the **no** form of this command.

debug platform hardware qfp active interface frame-relay multilink {*all*| *error*| *info*| *trace*| *warning*}

no debug platform hardware qfp active interface frame-relay multilink {*all*| *error*| *info*| *trace*| *warning*}

Syntax Description

mulitlink	Enables debug logging for the MFR multilink.
all	All debug levels**.**
error	**E** rror debug level**.**
info	**I** nformation debug level.
trace	**R** ace debug level.
warning	**W** arning debug level.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.4S	This command was introduced.

Examples

The following example shows how to debug the multilink frame relay client at all levels:

```
Router# debug platform hardware qfp active interface frame-relay multilink
all
The selected MFR Client debugging is on
```

debug rgf detailed

To enable detailed debugging information about redundancy group facility (RGF) events that are sent and received on Multirouter Automatic Protection Switching (MR-APS)-enabled routers that support stateful Multilink PPP (MLPPP) sessions, use the **debug rgf detailed** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug rgf detailed

no debug rgf detailed

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(3)S	This command was introduced.

Examples

The following is sample output from the **debug rgf detailed** command. The fields in the display are self-explanatory.

```
Router# debug rgf detailed
RGF detailed event debugging is on
6d00h: RGF: Rcvd aps evt[4] aps_group_id:1
6d00h: RGF Event: Group[1] Got event[Go-Standby-cold] current state[Standby-bulk]
6d00h: RGF: Group [1] state[Standby-bulk] Sending [Init] to client Id[1]
6d00h: RGF: Group[1] Client [1] Sent OK for Init
6d00h: RGF State: Group[1] Old State [Standby-bulk] New State [Init] Event [Go-Standby-cold]
6d00h: RGF: Group[1] buffer app data len[20] len[44] allocated
6d00h: RGF: Sending data group[1] client[0] app data len[20]
6d00h: RGF: Sending data dump
6d00h: ICRM HEADER:
 30 2 0 28
6d00h: RGF HEADER:
 0 0 0 2 0 0 0 14 0 0 0 1 0 0 0 0 0 0 0 0
6d00h: PAYLOAD:
 0 0 0 0 0 0 0 1 0 0 0 2 0 0 0 4 0 0 0 0
6d00h: RGF: Sent msg_id 43317, 44 bytes to ICRM conn_hdl0xAD000000
6d00h: RGF[1]: Client [1] Done for Init Action Going Cold
6d00h: RGF: Group [1] state[Init] Sending [Standby cold] to client Id[1]
6d00h: RGF[1]: Client [1] Done for Standby cold Action Going Bulk
6d00h: RGF State: Group[1] Old State [Init] New State [Standby-cold] Event [Go-Standby-cold]
6d00h: RGF: Group[1] buffer app data len[20] len[44] allocated
6d00h: RGF: Sending data group[1] client[0] app data len[20]
6d00h: RGF: Sending data dump
6d00h: ICRM HEADER:
 30 2 0 28
6d00h: RGF HEADER:
 0 0 0 2 0 0 0 14 0 0 0 1 0 0 0 0 0 0 0 0
6d00h: PAYLOAD:
 0 0 0 0 0 0 0 3 0 0 0 2 0 0 0 1 0 0 0 0
6d00h: RGF: Sent msg_id 43318, 44 bytes to ICRM conn_hdl0xAD000000
6d00h: RGF[1]: Dint get go bulk from APS. Postponing
```

Related Commands

Command	Description
debug rgf errors	Enables RGF error debugging.
debug rgf events	Displays debugging information of all RGF events.

debug rgf errors

To enable redundancy group facility (RGF) error debugging on Multirouter Automatic Protection Switching (MR-APS)-enabled routers that support stateful Multilink PPP (MLPPP) sessions, use the **debug rgf errors** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rgf errors

no debug rgf errors

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(3)S	This command was introduced.

Examples

The following example shows how to use this command to display any RGF errors that may have occurred in the system:

```
Router# debug rgf errors
RGF Error debugging is on
```

You will receive an error debugging output only if there are any RGF errors in the system.

Related Commands

Command	Description
debug rgf detailed	Displays detailed debugging information of RGF events sent and received on the router.
debug rgf events	Displays debugging information of all RGF events.

debug rgf events

To display all redundancy group facility (RGF) events on Multirouter Automatic Protection Switching (MR-APS)-enabled routers that support stateful Multilink PPP (MLPPP) sessions, use the **debug rgf events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rgf events

no debug rgf events

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(3)S	This command was introduced.

Examples

The following is sample output from the **debug rgf events** command when the SONET controller is shut. The fields in the display are self-explanatory:

```
Router# debug rgf events
RGF event debugging is on
Router#
6d00h: RGF: Rcvd aps evt[4] aps_group_id:1
6d00h: RGF[1]: Got Standby cold from APS. Wait for Peer
6d00h: RGF: Group[1] buffer app data len[20] len[44] allocated
6d00h: RGF: Sending data group[1] client[0] app data len[20]
6d00h: RGF: Sent msg_id 43218, 44 bytes to ICRM conn_hdl0xAD000000
6d00h: RGF: Rcvd aps evt[5] aps_group_id:1
6d00h: RGF PR PROG: Group[1] state [Standby-cold] Sending [peer Standby Bulk] to Peer
6d00h: RGF: Group[1] buffer app data len[20] len[44] allocated
6d00h: RGF: Sending data group[1] client[0] app data len[20]
6d00h: RGF: Sent msg_id 43315, 44 bytes to ICRM conn_hdl0xAD000000
6d00h: RGF State: Group[1] Old State [Standby-cold] New State [Standby-bulk] Event
[Go-Standby-bulk]
```

Related Commands

Command	Description
debug rgf detailed	Displays detailed debugging information of RGF events sent and received on the router.
debug rgf errors	Enables RGF error debugging.

debug vpdn

To troubleshoot Layer 2 Forwarding (L2F) or Layer 2 Tunnel Protocol (L2TP) virtual private dial-up network (VPDN) tunneling events and infrastructure, use the **debug vpdn** command in privileged EXEC mode. To disable the debugging of L2TP VPDN tunneling events and infrastructure, use the **no** form of this command.

Note Effective with Cisco IOS Release 12.4(11)T, the L2F protocol is not supported in Cisco IOS software.

Cisco IOS Release 12.2(33)XNA and Later Releases

debug vpdn {**call** {**event**| **fsm**}| **authorization** {**error**| **event**}| **error**| **event** [**disconnect** [**traceback**]]| **l2tp-sequencing**| **l2x-data**| **l2x-errors**| **l2x-events**| **l2x-packets**| **message**| **packet** [**detail**| **errors**]| **sss** {**error**| **event**| **fsm**}| **subscriber** {**error**| **event**| **fsm**}}

no debug vpdn {**call** {**event**| **fsm**}| **authorization** {**error**| **event**}| **error**| **event** [**disconnect** [**traceback**]]| **l2tp-sequencing**| **l2x-data**| **l2x-errors**| **l2x-events**| **l2x-packets**| **message**| **packet** [**detail**| **errors**]| **sss** {**error**| **event**| **fsm**}| **subscriber** {**error**| **event**| **fsm**}}

Cisco IOS Releases Prior to 12.2(33)XNA

debug vpdn {**call** {**event**| **fsm**}| **authorization** {**error**| **event**}| **error**| **event** [**disconnect**]| **l2tp-sequencing**| **l2x-data**| **l2x-errors**| **l2x-events**| **l2x-packets**| **message**| **packet** [**detail**| **errors**]| **sss** {**error**| **event**| **fsm**}| **subscriber** {**error**| **event**| **fsm**}}

no debug vpdn {**call** {**event**| **fsm**}| **authorization** {**error**| **event**}| **error**| **event** [**disconnect**]| **l2tp-sequencing**| **l2x-data**| **l2x-errors**| **l2x-events**| **l2x-packets**| **message**| **packet** [**detail**| **errors**]| **sss** {**error**| **event**| **fsm**}| **subscriber** {**error**| **event**| **fsm**}}

Syntax Description

call event	Displays significant events in the VPDN call manager.
call fsm	Displays significant events in the VPDN call manager finite state machine (FSM).
authorization error	Displays authorization errors.
authorization event	Displays authorization events.
error	Displays VPDN errors.
event	Displays VPDN events.
disconnect	(Optional) Displays VPDN disconnect events. **Note** The disconnect keyword is required in Cisco IOS Release 12.2(33)XNA and later releases.
traceback	(Optional) Displays traceback messages that provide reasons for VPDN disconnect.

l2tp-sequencing	Displays significant events related to L2TP sequence numbers such as mismatches, resend queue flushes, and drops.
l2x-data	Displays errors that occur in data packets.
l2x-errors	Displays errors that occur in protocol-specific conditions.
l2x-events	Displays events resulting from protocol-specific conditions.
l2x-packets	Displays detailed information about control packets in protocol-specific conditions.
message	Displays VPDN interprocess messages.
packet	Displays information about VPDN packets.
detail	(Optional) Displays detailed packet information, including packet dumps.
errors	(Optional) Displays errors that occur in packet processing.
sss error	Displays debug information about VPDN Subscriber Service Switch (SSS) errors.
sss event	Displays debug information about VPDN SSS events.
sss fsm	Displays debug information about the VPDN SSS FSM.
subscriber error	Displays debug information about VPDN Subscriber errors.
subscriber event	Displays debug information about VPDN Subscriber events.
subscriber fsm	Displays debug information about the VPDN Subscriber FSM.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
11.2 T	This command was introduced.

Release	Modification
12.0(5)T	This command was modified. Support was added for L2TP debugging messages. The **l2tp-sequencing** and **error** keywords were added. The **l2f-errors**, **l2f-events**, and **l2f-packets** keywords were changed to **l2x-errors**, **l2x-events**, and **l2x-packets**.
12.2(4)T	This command was modified. The **call**, **event**, **fsm**, and **message** keywords were added.
12.2(11)T	This command was modified. The **detail** keyword was added.
12.0(23)S	This command was integrated into Cisco IOS Release 12.0(23)S.
12.2(13)T	This command was modified. The **sss**, **error**, **event**, and **fsm** keywords were added.
12.3(14)T	This command was modified. Support was added to decode the outbound control channel authentication events.
12.0(31)S	This command was modified. The output was enhanced to display messages about control channel authentication events.
12.2(27)SBC	This command was modified. Support for enhanced display of messages about control channel authentication events was added.
12.2(28)SB	This command was modified. Support for the display of messages about congestion avoidance events was added.
12.2(31)SB	This command was modified. Support was added to decode the outbound control channel authentication events.
12.4(15)T	This command was modified. The **authorization**, **error**, and **event** keywords were added.
12.2(33)XNA	This command was modified. The **traceback** keyword was added.
12.4(20)T	This command was modified. The **subscriber** keyword was added and the **sss** keyword was removed.
Cisco IOS XE Release 2.6	This command was modified. Authentication failure messages for L2TPv3 were added.

Usage Guidelines

The **debug vpdn packet** and **debug vpdn packet detail** commands generate several debug operations per packet. Depending on the L2TP traffic pattern, these commands may cause the CPU load to increase to a high level that impacts performance.

Examples

Examples

The following example shows the VPDN configuration on a network access server (NAS):

```
vpdn-group 1
 request-dialin
  protocol l2f
  domain example.com
 initiate-to ip 172.17.33.125
username nas1 password nas1
```

The following is sample output from the **debug vpdn event** command on a NAS when an L2F tunnel is brought up and Challenge Handshake Authentication Protocol (CHAP) authentication of the tunnel succeeds:

```
Device# debug vpdn event

%LINK-3-UPDOWN: Interface Async6, changed state to up
*Mar 2 00:26:05.537: looking for tunnel — example.com —
*Mar 2 00:26:05.545: Async6 VPN Forwarding...
*Mar 2 00:26:05.545: Async6 VPN Bind interface direction=1
*Mar 2 00:26:05.553: Async6 VPN vpn_forward_user user6@example.com is forwarded
%LINEPROTO-5-UPDOWN: Line protocol on Interface Async6, changed state to up
*Mar 2 00:26:06.289: L2F: Chap authentication succeeded for nas1.
```

The following is sample output from the **debug vpdn event** command on a NAS when the L2F tunnel is brought down normally:

```
Device# debug vpdn event

%LINEPROTO-5-UPDOWN: Line protocol on Interface Async6, changed state to down
%LINK-5-CHANGED: Interface Async6, changed state to reset
*Mar 2 00:27:18.865: Async6 VPN cleanup
*Mar 2 00:27:18.869: Async6 VPN reset
*Mar 2 00:27:18.873: Async6 VPN Unbind interface
%LINK-3-UPDOWN: Interface Async6, changed state to down
```

The table below describes the significant fields shown in the two previous displays. The output describes normal operations when an L2F tunnel is brought up or down on a NAS.

Table 1: debug vpdn event Field Descriptions for the NAS

Field	Description
Asynchronous interface coming up	
%LINK-3-UPDOWN: Interface Async6, changed state to up	Asynchronous interface 6 came up.
looking for tunnel — example.com — Async6 VPN Forwarding...	Domain name is identified.

Field	Description
Async6 VPN Bind interface direction=1	Tunnel is bound to the interface. These are the direction values: • 1—From the NAS to the tunnel server • 2—From the tunnel server to the NAS
Async6 VPN vpn_forward_user user6@example.com is forwarded	Tunnel for the specified user and domain name is forwarded.
%LINEPROTO-5-UPDOWN: Line protocol on Interface Async6, changed state to up	Line protocol is up.
L2F: Chap authentication succeeded for nas1.	Tunnel was authenticated with the tunnel password nas1.
Virtual access interface coming down	
%LINEPROTO-5-UPDOWN: Line protocol on Interface Async6, changed state to down	Normal operation when the virtual access interface is taken down.
Async6 VPN cleanup Async6 VPN reset Async6 VPN Unbind interface	Normal cleanup operations performed when the line or virtual access interface goes down.

Examples

The following example shows the VPDN configuration on a tunnel server, which uses *nas1* as the tunnel name and the tunnel authentication name. The tunnel authentication name can be entered in a user's file on an authentication, authorization, and accounting (AAA) server and used to define authentication requirements for the tunnel.

```
vpdn-group 1
 accept-dialin
  protocol l2f
  virtual-template 1
 terminate-from hostname nas1
```

The following is sample output from the **debug vpdn event** command on a tunnel server when an L2F tunnel is brought up successfully:

```
Device# debug vpdn event

L2F: Chap authentication succeeded for nas1.
Virtual-Access3 VPN Virtual interface created for user6@example.com
Virtual-Access3 VPN Set to Async interface
Virtual-Access3 VPN Clone from Vtemplate 1 block=1 filterPPP=0
%LINK-3-UPDOWN: Interface Virtual-Access3, changed state to up
Virtual-Access3 VPN Bind interface direction=2
Virtual-Access3 VPN PPP LCP accepted sent & rcv CONFACK
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access3, changed state to up
```

The following is sample output from the **debug vpdn event** command on a tunnel server when an L2F tunnel is brought down normally:

```
Device# debug vpdn event

%LINK-3-UPDOWN: Interface Virtual-Access3, changed state to down
Virtual-Access3 VPN cleanup
Virtual-Access3 VPN reset
Virtual-Access3 VPN Unbind interface
Virtual-Access3 VPN reset
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access3, changed state to down
```

The table below describes the fields shown in two previous outputs. The output describes normal operations when an L2F tunnel is brought up or down on a tunnel server.

Table 2: debug vpdn event Field Descriptions

Field	Description
L2F: Chap authentication succeeded for nas1.	PPP CHAP authentication status for the tunnel named *nas1*.
Virtual-Access3 VPN Virtual interface created for user6@example.com	Virtual access interface was set up on a tunnel server for the user user6@example.com.
Virtual-Access3 VPN Set to Async interface	Virtual access interface 3 was set to asynchronous for character-by-character transmission.
Virtual-Access3 VPN Clone from Vtemplate 1 block=1 filterPPP=0	Virtual template 1 was applied to virtual access interface 3.
%LINK-3-UPDOWN: Interface Virtual-Access3, changed state to up	Link status is set to up.
Virtual-Access3 VPN Bind interface direction=2	Tunnel is bound to the interface. These are the direction values: • 1—From the NAS to the tunnel server • 2—From the tunnel server to the NAS
Virtual-Access3 VPN PPP LCP accepted sent & rcv CONFACK	PPP link control protocol (LCP) configuration settings (negotiated between the remote client and the NAS) were copied to the tunnel server and acknowledged.
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access3, changed state to up	Line protocol is up; the line can be used.
%LINK-3-UPDOWN: Interface Virtual-Access3, changed state to down	Virtual access interface is coming down.

Field	Description
Virtual-Access3 VPN cleanup Virtual-Access3 VPN reset Virtual-Access3 VPN Unbind interface Virtual-Access3 VPN reset	Device is performing normal cleanup operations when a virtual access interface used for an L2F tunnel comes down.
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access3, changed state to down	Line protocol is down for virtual access interface 3; the line cannot be used.

Examples

The following is sample output from the **debug vpdn event disconnect traceback** command on a tunnel server when an L2TP Network Server (LNS) tunnel session is disconnected:

```
Device# debug vpdn event disconnect traceback

*Aug  8 07:13:56.795: VPDN Vi2.1 disconnect (L2X) IETF: 18/host-request Ascend: 66/VPDN
Local PPP Disconnect
*Aug  8 07:13:56.795: VPDN Vi2.1 vpdn shutdown session, result=2, error=6, vendor_err=0,
syslog_error_code=2, syslog_key_type=1
*Aug  8 07:13:56.795: VPDN Vi2.1 VPDN/AAA: accounting stop sent
*Aug  8 07:13:56.795: VPDN Vi2.1 Unbinding session from idb, informational traceback:
*Aug  8 07:13:56.795: -Traceback= DFFFE7z 30EE221z 30DFBA8z 30E2F26z 30DF1DCz 30DF12Fz
1F0170Fz 1F015A1z 31E695Bz 31E674Dz 1F019F6z
*Aug  8 07:13:56.795: Vi2.1 VPDN: Resetting interface, informational traceback below:
LNS#
*Aug  8 07:13:56.795: -Traceback= DFFFE7z 30EDE74z 30EE2D4z 37996B7z 37A3019z 30EE408z
30DFBB3z 30E2F26z 30DF1DCz 30DF12Fz 1F0170Fz 1F015A1z 31E695Bz 31E674Dz 1F019F6z
```

Examples

The following is sample output from the **debug vpdn event** command on the NAS when an L2TP tunnel is brought up successfully:

```
Device# debug vpdn event

20:19:17: L2TP: I SCCRQ from ts1 tnl 8
20:19:17: L2X: Never heard of ts1
20:19:17: Tnl 7 L2TP: New tunnel created for remote ts1, address 172.21.9.4
20:19:17: Tnl 7 L2TP: Got a challenge in SCCRQ, ts1
20:19:17: Tnl 7 L2TP: Tunnel state change from idle to wait-ctl-reply
20:19:17: Tnl 7 L2TP: Got a Challenge Response in SCCCN from ts1
20:19:17: Tnl 7 L2TP: Tunnel Authentication success
20:19:17: Tnl 7 L2TP: Tunnel state change from wait-ctl-reply to established
20:19:17: Tnl 7 L2TP: SM State established
20:19:17: Tnl/Cl 7/1 L2TP: Session FS enabled
20:19:17: Tnl/Cl 7/1 L2TP: Session state change from idle to wait-for-tunnel
20:19:17: Tnl/Cl 7/1 L2TP: New session created
20:19:17: Tnl/Cl 7/1 L2TP: O ICRP to ts1 8/1
20:19:17: Tnl/Cl 7/1 L2TP: Session state change from wait-for-tunnel to wait-connect
20:19:17: Tnl/Cl 7/1 L2TP: Session state change from wait-connect to established
20:19:17: Vi1 VPDN: Virtual interface created for example1@example.com
20:19:17: Vi1 VPDN: Set to Async interface
20:19:17: Vi1 VPDN: Clone from Vtemplate 1 filterPPP=0 blocking
20:19:18: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
20:19:18: Vi1 VPDN: Bind interface direction=2
20:19:18: Vi1 VPDN: PPP LCP accepting rcv CONFACK
20:19:19: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
```

Examples

The following is sample output from the **debug vpdn event** command on a tunnel server when an L2TP tunnel is brought up successfully:

```
Device# debug vpdn event

20:47:33: %LINK-3-UPDOWN: Interface Async7, changed state to up
20:47:35: As7 VPDN: Looking for tunnel — example.com —
20:47:35: As7 VPDN: Get tunnel info for example.com with NAS nas1, IP 172.21.9.13
20:47:35: As7 VPDN: Forward to address 172.21.9.13
20:47:35: As7 VPDN: Forwarding...
20:47:35: As7 VPDN: Bind interface direction=1
20:47:35: Tnl/Cl 8/1 L2TP: Session FS enabled
20:47:35: Tnl/Cl 8/1 L2TP: Session state change from idle to wait-for-tunnel
20:47:35: As7 8/1 L2TP: Create session
20:47:35: Tnl 8 L2TP: SM State idle
20:47:35: Tnl 8 L2TP: Tunnel state change from idle to wait-ctl-reply
20:47:35: Tnl 8 L2TP: SM State wait-ctl-reply
20:47:35: As7 VPDN: example1@example.com is forwarded
20:47:35: Tnl 8 L2TP: Got a challenge from remote peer, nas1
20:47:35: Tnl 8 L2TP: Got a response from remote peer, nas1
20:47:35: Tnl 8 L2TP: Tunnel Authentication success
20:47:35: Tnl 8 L2TP: Tunnel state change from wait-ctl-reply to established
20:47:35: Tnl 8 L2TP: SM State established
20:47:35: As7 8/1 L2TP: Session state change from wait-for-tunnel to wait-reply
20:47:35: As7 8/1 L2TP: Session state change from wait-reply to established
20:47:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async7, changed state to up
```

Examples

The following is sample output from the **debug vpdn l2x-events** command on the NAS when an L2F tunnel is brought up successfully:

```
Device# debug vpdn l2x-events

%LINK-3-UPDOWN: Interface Async6, changed state to up
*Mar 2 00:41:17.365: L2F Open UDP socket to 172.21.9.26
*Mar 2 00:41:17.385: L2F_CONF received
*Mar 2 00:41:17.389: L2F Removing resend packet (type 1)
*Mar 2 00:41:17.477: L2F_OPEN received
*Mar 2 00:41:17.489: L2F Removing resend packet (type 2)
*Mar 2 00:41:17.493: L2F building nas2gw_mid0
%LINEPROTO-5-UPDOWN: Line protocol on Interface Async6, changed state to up
*Mar 2 00:41:18.613: L2F_OPEN received
*Mar 2 00:41:18.625: L2F Got a MID management packet
*Mar 2 00:41:18.625: L2F Removing resend packet (type 2)
*Mar 2 00:41:18.629: L2F MID synced NAS/HG Clid=7/15 Mid=1 on Async6
```

The following is sample output from the **debug vpdn l2x-events** command on a NAS when an L2F tunnel is brought down normally:

```
Device# debug vpdn l2x-events

%LINEPROTO-5-UPDOWN: Line protocol on Interface Async6, changed state to down
%LINK-5-CHANGED: Interface Async6, changed state to reset
*Mar 2 00:42:29.213: L2F_CLOSE received
*Mar 2 00:42:29.217: L2F Destroying mid
*Mar 2 00:42:29.217: L2F Removing resend packet (type 3)
*Mar 2 00:42:29.221: L2F Tunnel is going down!
*Mar 2 00:42:29.221: L2F Initiating tunnel shutdown.
*Mar 2 00:42:29.225: L2F_CLOSE received
*Mar 2 00:42:29.229: L2F_CLOSE received
*Mar 2 00:42:29.229: L2F Got closing for tunnel
*Mar 2 00:42:29.233: L2F Removing resend packet
*Mar 2 00:42:29.233: L2F Closed tunnel structure
%LINK-3-UPDOWN: Interface Async6, changed state to down
*Mar 2 00:42:31.793: L2F Closed tunnel structure
*Mar 2 00:42:31.793: L2F Deleted inactive tunnel
```

The table below describes the fields shown in the displays.

Table 3: debug vpdn l2x-events Field Descriptions—NAS

Field	Descriptions
%LINK-3-UPDOWN: Interface Async6, changed state to up	Asynchronous interface came up normally.
L2F Open UDP socket to 172.21.9.26	L2F opened a User Datagram Protocol (UDP) socket to the tunnel server IP address.
L2F_CONF received	L2F_CONF signal was received. When sent from the tunnel server to the NAS, an L2F_CONF indicates the tunnel server's recognition of the tunnel creation request.
L2F Removing resend packet (type ...)	Removing the resend packet for the L2F management packet. There are two resend packets that have different meanings in different states of the tunnel.
L2F_OPEN received	L2F_OPEN management message was received, indicating that the tunnel server accepted the NAS configuration of an L2F tunnel.
L2F building nas2gw_mid0	L2F is building a tunnel between the NAS and the tunnel server using the multiplex ID (MID) MID0.
%LINEPROTO-5-UPDOWN: Line protocol on Interface Async6, changed state to up	Line protocol came up. Indicates whether the software processes that handle the line protocol regard the interface as usable.
L2F_OPEN received	L2F_OPEN management message was received, indicating that the tunnel server accepted the NAS configuration of an L2F tunnel.
L2F Got a MID management packet	MID management packets are used to communicate between the NAS and the tunnel server.
L2F MID synced NAS/HG Clid=7/15 Mid=1 on Async6	L2F synchronized the client IDs on the NAS and the tunnel server, respectively. An MID is assigned to identify this connection in the tunnel.
Tunnel coming down	
%LINEPROTO-5-UPDOWN: Line protocol on Interface Async6, changed state to down	Line protocol came down. Indicates whether the software processes that handle the line protocol regard the interface as usable.

Field	Descriptions
%LINK-5-CHANGED: Interface Async6, changed state to reset	Interface was marked as reset.
L2F_CLOSE received	NAS received a request to close the tunnel.
L2F Destroying mid	Connection identified by the MID is being taken down.
L2F Tunnel is going down!	Advisory message about impending tunnel shutdown.
L2F Initiating tunnel shutdown.	Tunnel shutdown has started.
L2F_CLOSE received	NAS received a request to close the tunnel.
L2F Got closing for tunnel	NAS began tunnel closing operations.
%LINK-3-UPDOWN: Interface Async6, changed state to down	Asynchronous interface was taken down.
L2F Closed tunnel structure	NAS closed the tunnel.
L2F Deleted inactive tunnel	Now-inactivated tunnel was deleted.

Examples

The following is sample output from the **debug vpdn l2x-events** command on a tunnel server when an L2F tunnel is created:

```
Device# debug vpdn l2x-events

L2F_CONF received
L2F Creating new tunnel for nas1
L2F Got a tunnel named nas1, responding
L2F Open UDP socket to 172.21.9.25
L2F_OPEN received
L2F Removing resend packet (type 1)
L2F_OPEN received
L2F Got a MID management packet
%LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
```

The following is sample output from the **debug vpdn l2x-events** command on a tunnel server when the L2F tunnel is brought down normally:

```
Device# debug vpdn l2x-events

L2F_CLOSE received
L2F Destroying mid
L2F Removing resend packet (type 3)
L2F Tunnel is going down!
L2F Initiating tunnel shutdown.
%LINK-3-UPDOWN: Interface Virtual-Access1, changed state to down
L2F_CLOSE received
L2F Got closing for tunnel
L2F Removing resend packet
L2F Removing resend packet
L2F Closed tunnel structure
```

```
L2F Closed tunnel structure
L2F Deleted inactive tunnel
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to down
```

The table below describes the significant fields shown in the displays.

Table 4: debug vpdn l2x-events Field Descriptions—Tunnel Server

Field	Description
L2F_CONF received	L2F configuration is received from the NAS. When sent from a NAS to a tunnel server, the L2F_CONF is the initial packet in the conversation.
L2F Creating new tunnel for nas1	Tunnel named nas1 is being created.
L2F Got a tunnel named nas1, responding	Tunnel server is responding.
L2F Open UDP socket to 172.21.9.25	Opening a socket to the NAS IP address.
L2F_OPEN received	L2F_OPEN management message was received, indicating that the NAS is opening an L2F tunnel.
L2F Removing resend packet (type 1)	Removing the resend packet for the L2F management packet. The two resend packet types have different meanings in different states of the tunnel.
L2F Got a MID management packet	L2F MID management packets are used to communicate between the NAS and the tunnel server.
%LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up	Tunnel server is bringing up virtual access interface 1 for the L2F tunnel.
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up	Line protocol is up. The line can be used.
Tunnel coming down	
L2F_CLOSE received	NAS or tunnel server received a request to close the tunnel.
L2F Destroying mid	Connection identified by the MID is being taken down.
L2F Removing resend packet (type 3)	Removing the resend packet for the L2F management packet. There are two resend packets that have different meanings in different states of the tunnel.
L2F Tunnel is going down! L2F Initiating tunnel shutdown.	Device is performing normal operations when a tunnel is coming down.

Field	Description
%LINK-3-UPDOWN: Interface Virtual-Access1, changed state to down	The virtual access interface is coming down.
L2F_CLOSE received L2F Got closing for tunnel L2F Removing resend packet L2F Removing resend packet L2F Closed tunnel structure L2F Closed tunnel structure L2F Deleted inactive tunnel	Device is performing normal cleanup operations when the tunnel is being brought down.
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to down	Line protocol is down; virtual access interface 1 cannot be used.

Examples

The following partial example of the **debug vpdn l2x-events** command is useful for monitoring a network running the L2TP Congestion Avoidance feature. The report shows that the congestion window (Cwnd) has been reset to 1 because of packet retransmissions:

```
Device# debug vpdn l2x-events
.
.
.
*Jul 15 19:02:57.963:  Tnl 47100 L2TP: Congestion Control event received is retransmission
*Jul 15 19:02:57.963:  Tnl 47100 L2TP: Congestion Window size, Cwnd 1
*Jul 15 19:02:57.963:  Tnl 47100 L2TP: Slow Start threshold, Ssthresh 2
*Jul 15 19:02:57.963:  Tnl 47100 L2TP: Remote Window size, 500
*Jul 15 19:02:57.963:  Tnl 47100 L2TP: Control channel retransmit delay set to 4 seconds
*Jul 15 19:03:01.607:  Tnl 47100 L2TP: Update ns/nr, peer ns/nr 2/5, our ns/nr 5/2
```

The following partial example shows that traffic has been restarted with L2TP congestion avoidance throttling traffic:

```
Device# debug vpdn l2x-events
.
.
.
*Jul 15 14:45:16.123:  Tnl 30597 L2TP: Control channel retransmit delay set to 2 seconds
*Jul 15 14:45:16.123:  Tnl 30597 L2TP: Tunnel state change from idle to wait-ctl-reply
*Jul 15 14:45:16.131:  Tnl 30597 L2TP: Congestion Control event received is positive
acknowledgement
*Jul 15 14:45:16.131:  Tnl 30597 L2TP: Congestion Window size, Cwnd 2
*Jul 15 14:45:16.131:  Tnl 30597 L2TP: Slow Start threshold, Ssthresh 500
*Jul 15 14:45:16.131:  Tnl 30597 L2TP: Remote Window size, 500
*Jul 15 14:45:16.131:  Tnl 30597 L2TP: Congestion Ctrl Mode is Slow Start
```

The table below describes the significant fields shown in the displays. See RFC 2661 for more details about the information in the reports for L2TP congestion avoidance.

Table 5: debug vpdn l2x-events Field Descriptions—L2TP Congestion Avoidance

Field	Description
Control channel retransmit delay set to ...	Indicates the current value set for the retransmit delay.
Tunnel state...	Indicates the tunnel's current Control Connection State, per RFC 2661.
Congestion Control event received is...	Indicates the received congestion control event. • Retransmission—Indicates packet retransmission has been detected in the resend queue. • Positive acknowledgement—Indicates that a packet was received and acknowledged by the peer tunnel endpoint.
Congestion Window size, Cwnd 2	Current size of the Cwnd.
Slow Start threshold, Ssthresh 500	Current value of the slow start threshold (Ssthresh).
Remote Window size, 500	Size of the advertised receive window configured on the remote peer with the **l2tp tunnel receive-window** command.
Congestion Ctrl Mode is...	Indicates whether the device is operating in Slow Start or Congestion Avoidance mode.
Update ns/nr, peer ns/nr 2/5, our ns/nr 5/2	See RFC 2661.

Examples

The following is sample output from the **debug vpdn error** command on a NAS when the L2F tunnel is not set up:

```
Device# debug vpdn error

%LINEPROTO-5-UPDOWN: Line protocol on Interface Async1, changed state to down
%LINK-5-CHANGED: Interface Async1, changed state to reset
%LINK-3-UPDOWN: Interface Async1, changed state to down
%LINK-3-UPDOWN: Interface Async1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Async1, changed state to up
VPDN tunnel management packet failed to authenticate
VPDN tunnel management packet failed to authenticate
```

The table below describes the significant fields shown in the display.

Table 6: debug vpdn error Field Descriptions for the NAS

Field	Description
%LINEPROTO-5-UPDOWN: Line protocol on Interface Async1, changed state to down	Line protocol on the asynchronous interface went down.
%LINK-5-CHANGED: Interface Async1, changed state to reset	Asynchronous interface 1 was reset.
%LINK-3-UPDOWN: Interface Async1, changed state to down %LINK-3-UPDOWN: Interface Async1, changed state to up	Link from asynchronous interface 1 link went down and then came back up.
%LINEPROTO-5-UPDOWN: Line protocol on Interface Async1, changed state to up	Line protocol on the asynchronous interface came back up.
VPDN tunnel management packet failed to authenticate	Tunnel authentication failed. This is the most common VPDN error. **Note** Verify the password for the NAS and the tunnel server name. If you store the password on an AAA server, you can use the **debug aaa authentication** command.

The following is sample output from the **debug vpdn l2x-errors** command:

```
Device# debug vpdn l2x-errors

%LINK-3-UPDOWN: Interface Async1, changed state to up
L2F Out of sequence packet 0 (expecting 0)
L2F Tunnel authentication succeeded for example.com
 L2F Received a close request for a non-existent mid
 L2F Out of sequence packet 0 (expecting 0)
 L2F packet has bogus1 key 1020868 D248BA0F
L2F packet has bogus1 key 1020868 D248BA0F
```

The table below describes the significant fields shown in the display.

Table 7: debug vpdn l2x-errors Field Descriptions

Field	Description
%LINK-3-UPDOWN: Interface Async1, changed state to up	The line protocol on the asynchronous interface came up.
L2F Out of sequence packet 0 (expecting 0)	Packet was expected to be the first in a sequence starting at 0, but an invalid sequence number was received.
L2F Tunnel authentication succeeded for example.com	Tunnel was established from the NAS to the tunnel server, example.com.

Field	Description
L2F Received a close request for a non-existent mid	Multiplex ID was not used previously; cannot close the tunnel.
L2F Out of sequence packet 0 (expecting 0)	Packet was expected to be the first in a sequence starting at 0, but an invalid sequence number was received.
L2F packet has bogus1 key 1020868 D248BA0F	Value based on the authentication response given to the peer during tunnel creation. This packet, in which the key does not match the expected value, must be discarded.
L2F packet has bogus1 key 1020868 D248BA0F	Another packet was received with an invalid key value. The packet must be discarded.

Examples

The following is sample output from the **debug vpdn l2x-packets** command on a NAS. This example displays a trace for a **ping** command.

```
Device# debug vpdn l2x-packets

L2F SENDING (17): D0 1 1 10 0 0 0 4 0 11 0 0 81 94 E1 A0 4
L2F header flags: 53249 version 53249 protocol 1 sequence 16 mid 0 cid 4
length 17 offset 0 key 1701976070
L2F RECEIVED (17): D0 1 1 10 0 0 0 4 0 11 0 0 65 72 18 6 5
L2F SENDING (17): D0 1 1 11 0 0 0 4 0 11 0 0 81 94 E1 A0 4
L2F header flags: 53249 version 53249 protocol 1 sequence 17 mid 0 cid 4
length 17 offset 0 key 1701976070
L2F RECEIVED (17): D0 1 1 11 0 0 0 4 0 11 0 0 65 72 18 6 5
L2F header flags: 57345 version 57345 protocol 2 sequence 0 mid 1 cid 4
length 32 offset 0 key 1701976070
L2F-IN Output to Async1 (16): FF 3 C0 21 9 F 0 C 0 1D 41 AD FF 11 46 87
L2F-OUT (16): FF 3 C0 21 A F 0 C 0 1A C9 BD FF 11 46 87
L2F header flags: 49153 version 49153 protocol 2 sequence 0 mid 1 cid 4
length 32 offset 0 key -2120949344
L2F-OUT (101): 21 45 0 0 64 0 10 0 0 FF 1 B9 85 1 0 0 3 1 0 0 1 8 0 62 B1
0 0 C A8 0 0 0 0 0 0 11 E E0 AB CD AB CD AB CD AB CD AB CD AB CD AB CD
AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD
CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB
L2F header flags: 49153 version 49153 protocol 2 sequence 0 mid 1 cid 4
length 120 offset 3 key -2120949344
L2F header flags: 49153 version 49153 protocol 2 sequence 0 mid 1 cid 4
length 120 offset 3 key 1701976070
L2F-IN Output to Async1 (101): 21 45 0 0 64 0 10 0 0 FF 1 B9 85 1 0 0 1 1 0
0 3 0 0 6A B1 0 0 C A8 0 0 0 0 0 11 E E0 AB CD AB CD AB CD AB CD AB CD AB CD
AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB
CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD
```

The table below describes the significant fields shown in the display.

Table 8: debug vpdn l2x-packets Field Descriptions

Field	Description
L2F SENDING (17)	Number of bytes being sent. The first set of "SENDING"..."RECEIVED" lines displays L2F keepalive traffic. The second set displays L2F management data.
L2F header flags:	Version and flags, in decimal.
version 53249	Version number.
protocol 1	Protocol for negotiation of the point-to-point link between the NAS and the tunnel server is always 1, indicating L2F management.
sequence 16	Sequence numbers start at 0. Each subsequent packet is sent with the next increment of the sequence number. The sequence number is thus a free running counter represented modulo 256. There is a distinct sequence counter for each distinct MID value.
mid 0	MID, which identifies a particular connection within the tunnel. Each new connection is assigned a MID currently unused within the tunnel.
cid 4	Client ID used to assist endpoints in demultiplexing tunnels.
length 17	Size in octets of the entire packet, including header, all fields pre-sent, and payload. Length does not reflect the addition of the checksum, if present.
offset 0	Number of bytes past the L2F header at which the payload data is expected to start. If it is 0, the first byte following the last byte of the L2F header is the first byte of payload data.
key 1701976070	Value based on the authentication response given to the peer during tunnel creation. During the life of a session, the key value serves to resist attacks based on spoofing. If a packet is received in which the key does not match the expected value, the packet must be silently discarded.
L2F RECEIVED (17)	Number of bytes received.
L2F-IN Output to Async1 (16)	Payload datagram. The data came in to the VPDN code.

Field	Description
L2F-OUT (16):	Payload datagram sent out from the VPDN code to the tunnel.
L2F-OUT (101)	Ping payload datagram. The value 62 in this line is the ping packet size in hexadecimal (98 in decimal). The three lines that follow this line show ping packet data.

Examples

The following example shows output from the **debug vpdn l2x-events** command for an L2TP version 3 (L2TPv3) xconnect session on an Ethernet interface:

```
Device# debug vpdn l2x-events

23:31:18: L2X: l2tun session [1669204400], event [client request], old state [open], new
state [open]
 23:31:18: L2X: L2TP: Received L2TUN message <Connect>
 23:31:18: Tnl/Sn58458/28568 L2TP: Session state change from idle to wait-for-tunnel
 23:31:18: Tnl/Sn58458/28568 L2TP: Create session
 23:31:18: Tnl58458 L2TP: SM State idle
 23:31:18: Tnl58458 L2TP: O SCCRQ
 23:31:18: Tnl58458 L2TP: Control channel retransmit delay set to 1 seconds
 23:31:18: Tnl58458 L2TP: Tunnel state change from idle to wait-ctl-reply
 23:31:18: Tnl58458 L2TP: SM State wait-ctl-reply
 23:31:18: Tnl58458 L2TP: I SCCRP from router
 23:31:18: Tnl58458 L2TP: Tunnel state change from wait-ctl-reply to established
 23:31:18: Tnl58458 L2TP: O SCCCN to router tnlid 8012
 23:31:18: Tnl58458 L2TP: Control channel retransmit delay set to 1 seconds
 23:31:18: Tnl58458 L2TP: SM State established
 23:31:18: Tnl/Sn58458/28568 L2TP: O ICRQ to router 8012/0
 23:31:18: Tnl/Sn58458/28568 L2TP: Session state change from wait-for-tunnel to wait-reply

 23:31:19: Tnl58458 L2TP: Control channel retransmit delay set to 1 seconds
 23:31:20: %LINK-3-UPDOWN: Interface Ethernet2/1, changed state to up
 23:31:21: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet2/1, changed state to
up
 23:31:25: L2X: Sending L2TUN message <Connect OK>
 23:31:25: Tnl/Sn58458/28568 L2TP: O ICCN to router 8012/35149
 23:31:25: Tnl58458 L2TP: Control channel retransmit delay set to 1 seconds
 23:31:25: Tnl/Sn58458/28568 L2TP: Session state change from wait-reply to established
 23:31:25: L2X: l2tun session [1669204400], event [server response], old state [open], new
 state [open]
 23:31:26: Tnl58458 L2TP: Control channel retransmit delay set to 1 seconds
```

Examples

The following example shows debug messages for control channel authentication failure events in Cisco IOS Release 12.0(31)S:

```
Device# debug vpdn l2x-events

Tnl41855 L2TP: Per-Tunnel auth counter, Overall Failed, now 1
Tnl41855 L2TP: Tunnel auth counter, Overall Failed, now 219
```

Related Commands

Command	Description
debug aaa authentication	Displays information on AAA/TACACS+ authentication.
debug acircuit	Displays events and failures related to attachment circuits.
debug pppoe	Displays debugging information for PPPoE sessions.
debug vpdn pppoe-data	Displays data packets of PPPoE sessions.
debug vpdn pppoe-error	Displays PPPoE protocol errors that prevent a session from being established or errors that cause an established sessions to be closed.
debug vpdn pppoe-events	Displays PPPoE protocol messages about events that are part of normal session establishment or shutdown.
debug vpdn pppoe-packet	Displays each PPPoE protocol packet exchanged.
debug xconnect	Displays errors and events related to an xconnect configuration.

debug waas

To enable debugging for WAAS Express modules, use the **debug waas** command in privileged EXEC mode. To disable WAAS Express debugging, use the **no** form of this command.

debug waas {{**auto-discovery**| **aoim**| **cce**| **infrastructure**| **lz**| **memory**| **tfo**} {**events**| **errors**| **operations**}| **api**| **mibs**| **dre** {**events**| **errors**| **operations [brief]**| **uplink**}| **management** {**events**| **errors**}}

no debug waas {{**auto-discovery**| **aoim**| **cce**| **infrastructure**| **lz**| **memory**| **tfo**} {**events**| **errors**| **operations**}| **api**| **mibs**| **dre** {**events**| **errors**| **operations [brief]**| **uplink**}| **management** {**events**| **errors**}}

Syntax Description

auto-discovery	Enables debugging for WAAS Express autodiscovery information.
aoim	Enables debugging for peer information and negotiated capabilities information.
cce	Enables debugging for Common Classification Engine (CCE).
infrastructure	Enables debugging for WAAS Express infrastructure.
lz	Enables debugging for Lempel-Ziv (LZ) optimization.
memory	Enables debugging for WAAS Express internal memory usage.
tfo	Enables debugging for Transport Flow Optimization (TFO).
events	Enables debugging for WAAS Express events.
errors	Enables debugging for WAAS Express errors.
operations	Enables debugging for WAAS Express operations.
brief	Displays WAAS connection operations in brief.
api	Enables debugging for WAAS Express public application programming interfaces (APIs).
mibs	Enables debugging for WAAS Express MIBs.
dre	Enables debugging for Data Redundancy Elimination (DRE) optimization.
uplink	Enables debugging for DRE upload.
management	Enables debugging for error and event management.

Command Default

Debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(2)T	This command was introduced.
15.2(3)T	This command was modified. The **api** and **mibs** keywords were added, and the **brief** keyword was removed.

Examples

The following example shows how to enable debugging output for WAAS Express infrastructure operations:

```
Device> enable
Device# debug waas infrastructure operations
```

Related Commands

Command	Description
clear waas	Clears WAAS Express statistics and closed connections information.
show waas alarms	Displays WAAS Express status and alarms.
show waas auto-discovery	Displays information about WAAS Express autodiscovery.
show waas connection	Displays information about WAAS Express connections.
show waas statistics aoim	Displays WAAS Express peer information and negotiated capabilities.
show waas statistics application	Displays WAAS Express policy application statistics.
show waas statistics auto-discovery	Displays WAAS Express autodiscovery statistics.
show waas statistics class	Displays statistics for the WAAS Express class map.
show waas statistics dre	Displays WAAS Express DRE statistics.
show waas statistics errors	Displays WAAS Express error statistics.
show waas statistics global	Displays global WAAS Express statistics.
show waas statistics lz	Displays WAAS Express LZ statistics.
show waas statistics pass-through	Displays WAAS Express connections placed in a pass-through mode.

Command	Description
show waas statistics peer	Displays inbound and outbound statistics for peer WAAS Express devices.
show waas status	Displays the status of WAAS Express.
show waas token	Displays the value of the configuration token used by the WAAS Central Manager.
waas cm-register url	Registers a device with the WAAS Central Manager.

digest

To enable Layer 2 Tunneling Protocol Version 3 (L2TPv3) control channel authentication or integrity checking, use the **digest**command in L2TP class configuration mode. To disable control channel authentication or integrity checking, use the **no** form of this command.

digest [**secret** [**0**| **7**] *password*] [**hash** {**md5**| **sha**}]

no digest [**secret** [**0**| **7**] *password* [**hash** {**md5**| **sha**}]]

Syntax Description

secret	(Optional) Enables L2TPv3 control channel authentication. If the **digest** command is issued without the **secret** keyword option, L2TPv3 integrity checking will be enabled.
[**0** | **7**]	Specifies the input format of the shared secret. • **0** --Specifies that a plain-text secret will be entered. • **7** --Specifies that an encrypted secret will be entered. The default value is **0**.
password	The shared secret used between peer provider edge (PE) routers. The value entered for the *password* argument must be in the format that matches the input format specified by the [**0** | **7**] keyword option.
hash {**md5**| **sha**}	(Optional) Specifies the hash function to be used in per-message digest calculations. • **md5** --Specifies HMAC-MD5 hashing. • **sha** --Specifies HMAC-SHA-1 hashing. The default hash function is **md5**.

Command Default

L2TPv3 control channel authentication and integrity checking are disabled by default.

Command Modes

L2TP class configuration

Command History

Release	Modification
12.0(29)S	This command was introduced.

Release	Modification
12.0(30)S	This command was enhanced to allow two different passwords to be configured simultaneously.
12.2(27)SBC	Support for this command was integrated into Cisco IOS Release 12.2(27)SBC.

Usage Guidelines

Beginning in Cisco IOS Release 12.0(29)S, two methods of control channel authentication are available. The L2TPv3 Control Message Hashing feature (enabled with the **digest**command) introduces a more robust authentication method than the older Challenge Handshake Authentication Protocol (CHAP) style method of authentication enabled with the **authentication**command. You may choose to enable both methods of authentication to ensure interoperability with peers that support only one of these methods of authentication, but this configuration will yield control of which authentication method is used to the peer PE router. Enabling both methods of authentication should be considered an interim solution to solve backward-compatibility issues during software upgrades.

The table below shows a compatibility matrix for the different L2TPv3 authentication methods. PE1 is running a Cisco IOS software release that supports the L2TPv3 Control Message Hashing feature, and the different possible authentication configurations for PE1 are shown in the first column. Each remaining column represents PE2 running software with different available authentication options, and the intersections indicate the different compatible configuration options for PE2. If any PE1/PE2 authentication configuration poses ambiguity on which method of authentication will be used, the winning authentication method is indicated in bold. If both the old and new authentication methods are enabled on PE1 and PE2, both types of authentication will occur.

Table 9: Compatibility Matrix for L2TPv3 Authentication Methods

PE1 Authentication Configuration	PE2 Supporting Old Authentication[1]	PE2 Supporting New Authentication[2]	PE2 Supporting Old and New Authentication[3]
None	None	None New integrity check	None New integrity check
Old authentication	Old authentication	--	Old authentication **Old authentication** and new authentication **Old authentication** and new integrity check
New authentication	--	New authentication	New authentication Old authentication and **new authentication**
New integrity check	None	None New integrity check	None New integrity check

PE1 Authentication Configuration	PE2 Supporting Old Authentication[1]	PE2 Supporting New Authentication[2]	PE2 Supporting Old and New Authentication[3]
Old and new authentication	Old authentication	New authentication	Old authentication New authentication **Old and new authentication** **Old authentication** and new integrity check
Old authentication and new integrity check	Old authentication	--	Old authentication **Old authentication** and new authentication **Old authentication** and new integrity check

1 Any PE software that supports only the old CHAP-like authentication system.

2 Any PE software that supports only the new message digest authentication and integrity checking authentication system, but does not understand the old CHAP-like authentication system. This type of software may be implemented by other vendors based on the latest L2TPv3 draft.

3 Any PE software that supports both the old CHAP-like authentication and the new message digest authentication and integrity checking authentication system, such as Cisco IOS 12.0(29)S or later releases.

In Cisco IOS Release 12.0(30)S, this command was enhanced to allow two L2TPv3 control channel authentication passwords to be configured simultaneously. This enhancement allows the transition from using an old authentication password to using a new authentication password without interrupting L2TPv3 services. No more than two passwords may be configured at a time. In order to configure a new password when two passwords are already configured, you must remove one of the existing passwords using the **no digest secret***password* command. The number of configured passwords can be verified using the **show l2tun tunnel**command.

Examples

The following example configures control channel authentication and a control channel authentication password for tunnels belonging to the L2TP class named class1:

```
l2tp-class class1
 digest secret cisco hash sha
 hidden
```

The following example configures a second control channel authentication password for tunnels belonging to the L2TP class named class1:

```
l2tp-class class1
 digest secret cisco2 hash sha
```

The following example removes the old control channel authentication password for tunnels belonging to the L2TP class named class1. The old password should be removed only after all peer routers have been configured with the new password.

```
l2tp-class class1
 no digest secret cisco hash sha
```

The following example configures control channel integrity checking and disables validation of the message digest for L2TPv3 tunnels belonging to the L2TP class named class2:

```
l2tp-class class2
 digest hash sha
 no digest check
```

The following example disables validation of the message digest for L2TPv3 tunnels belonging to the L2TP class named class3. Control channel authentication and control channel integrity checking are both disabled.

```
l2tp-class class3
 no digest check
```

Related Commands

Command	Description
authentication	Enables L2TPv3 CHAP-style authentication.
digest check	Enables the validation of the message digest in received control messages.
l2tp class	Creates a template of L2TP control plane configuration settings that can be inherited by different pseudowire classes and enters L2TP class configuration mode.
show l2tun tunnel	Displays the current state of L2TPv3 tunnels and displays information about currently configured tunnels, including local and remote L2TP hostnames, aggregate packet counts, and L2TP control channels.

dre upload

To enable upload Data Redundancy Elimination (DRE), use the **dre upload** command in parameter map configuration mode. To disable upload DRE, use the **no** form of this command.

dre upload

no dre upload

Syntax Description

This command has no arguments or keywords.

Command Default

Upload DRE is enabled.

Command Modes

Parameter map configuration (config-profile)

Command History

Release	Modification
15.2(3)T	This command was introduced.

Usage Guidelines

Upload DRE compresses data in the upload direction. Upload DRE is useful in the download-edit-upload scenario, where a user in a branch office downloads a file from the data center, modifies the file, and uploads the modified document back to the data center. If the modifications are small and localized, the upload of the modified file can benefit from the unmodified contents stored in the DRE cache.

Upload DRE is enabled by default. You can disable upload DRE by using the **no dre upload** command for troubleshooting purposes, and then you can enable it again. Download DRE is always enabled and cannot be disabled.

Examples

The following example shows how to disable upload DRE:

```
Device(config)# parameter-map type waas waas_global
Device(config-profile)# no dre upload
```

Related Commands

Command	Description
parameter-map type waas	Configures WAAS Express global parameters.
show waas accelerator	Displays information about WAAS Express accelerators.
show waas statistics dre	Displays WAAS Express DRE statistics.

dre-hints enable

To enable HTTP-Express accelerator to send Data Redundancy Elimination (DRE) hints to the DRE module, use the **dre-hints enable** command in WAAS HTTP configuration mode. To disable DRE hints, use the **no** form of this command.

dre-hints enable

no dre-hints enable

Syntax Description

This command has no arguments or keywords.

Command Default

DRE hints are enabled.

Command Modes

WAAS HTTP configuration (config-waas-http)

Command History

Release	Modification
15.2(3)T	This command was introduced.

Usage Guidelines

HTTP-Express accelerator can pass DRE hints to the DRE module at any point during a session. DRE hints help to improve overall DRE efficiency.

HTTP-Express accelerator can provide the following useful hints to the DRE module:

- Apply Lempel-Ziv (LZ) or Not: When the response from the server is already compressed, such as in the form of a jpeg or gzip file, HTTP-Express accelerator can instruct the DRE module to not apply LZ compression again. This can save some CPU cycles on WAAS Express.
- Skip Bytes Multiple: Multiple HTTP requests that request for the same file can have different headers even if the file being transferred is the same. To improve DRE compression in these cases, HTTP-Express accelerator can instruct DRE to skip the header bytes.

Before you can enable the **dre-hints enable** command, use the following commands:

- Use the **parameter-map type waas** command in global configuration mode to enter parameter map configuration mode.
- Use the **accelerator http-express** command in parameter map configuration mode to enter WAAS HTTP configuration mode.

Examples

The following example shows how to enable DRE hints:

```
Device(config)# parameter-map type waas waas_global
Device(config-profile)# accelerator http-express
```

```
Device(config-waas-http)# enable
Device(config-waas-http)# dre-hints enable
```

Related Commands

Command	Description
accelerator	Enters a specific WAAS Express accelerator configuration mode based on the accelerator being configured.
parameter-map type waas	Configures WAAS Express global parameters.
show waas connection detailed	Displays WAAS Express connection details.
show waas statistics dre	Displays WAAS Express DRE statistics.

dscp (Frame Relay VC-bundle-member)

To configure the differentiated services code point (DSCP) levels for a Frame Relay permanent virtual circuit (PVC) bundle member, use the **dscp** command in Frame Relay VC-bundle-member configuration mode. To remove the DSCP level configuration from the PVC, use the **no** form of this command.

dscp {*level*| **other**}

no dscp *level*

Syntax Description

level	DSCP level or levels for the Frame Relay PVC bundle member. The range is from 0 to 63. A PVC bundle member can be configured with a single DSCP level, multiple individual DSCP levels, a range of DSCP levels, multiple ranges of DSCP levels, or a combination of individual levels and level ranges. For example: • 9 • 25,35,45 • 25-35,45-55 • 10,20,25-35,40,45-55,60
other	Specifies that the Frame Relay PVC bundle member will handle all of the remaining DSCP levels that are not specified by other PVC bundle members.

Command Default

DSCP levels are not configured.

Command Modes

Frame Relay VC-bundle-member configuration

Command History

Release	Modification
12.2(13)T	This command was introduced.
12.2(16)BX	This command was integrated into Cisco IOS Release 12.2(16)BX.
12.0(26)S	This command was integrated into Cisco IOS Release 12.0(26)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Release	Modification
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Assignment of DSCP levels to PVC bundle members lets you create differentiated service, because you can distribute the DSCP levels over the various PVC bundle members. You can map a single DSCP level or range of levels to each discrete PVC in the bundle, which enables PVCs in the bundle to carry packets marked with different DSCP levels.

Use the **dscp other** command to configure a PVC to carry traffic marked with DSCP levels not specifically configured on other PVCs. Only one PVC in the bundle can be configured with the **dscp other** command.

This command is available only when the match type for the PVC bundle is set to **dscp** by using the **match dscp** command in Frame Relay VC-bundle configuration mode.

You can overwrite the DSCP level configuration on a PVC by reentering the **dscp** command with a new level value.

There is no default value for this command. When the PVC bundle is set to **dscp** using the **match dscp** command, all PVCs in the bundle are reset to remove any existing DSCP values. If one or more DSCP values are not specifically configured, the bundle will not come up.

However, a PVC may exist in a bundle but have no DSCP value associated with the bundle. As long as all valid DSCP values are handled by one or more of the other PVCs in the bundle, the bundle can come up, but the PVC that has no DSCP value configured will not participate in the bundle.

A DSCP level can be configured on one PVC bundle member per bundle. If you configure the same DSCP level on more than one PVC within a bundle, the following error warning appears on the console:

```
%Overlapping diff-serv code points
```

Examples

The following example assigns DSCP levels 0 through 9 to PVC bundle member 300 in a Frame Relay PVC bundle named MP-3-static:

interface Serial4/0

encapsulation frame-relay

frame-relay vc-bundle MP-3-static

match dscp

pvc 300

dscp 0-9

```
 frame-relay map ip 10.2.2.2 vc-bundle MP-3-static
```

The following example changes the DSCP levels in the above example from 0 through 9 to 0, 9, and 20 through 29:

```
interface serial 1/4
 frame-relay map ip 10.2.2.2 vc-bundle MP-3-static
 frame-relay vc-bundle MP-3-static
  match dscp
```

```
pvc 300
 dscp 0,9,20-29
```

Related Commands

Command	Description
exp	Configures MPLS EXP levels for a Frame Relay PVC bundle member.
frame-relay map	Defines mapping between a destination protocol address and the DLCI used to connect to the destination address.
frame-relay vc-bundle	Creates a Frame Relay PVC bundle and enters Frame Relay VC-bundle configuration mode.
match	Specifies which bits in the ToS octet to use for mapping packet service levels to Frame Relay PVC bundle members.
precedence (Frame Relay VC-bundle-member)	Configures the precedence levels for a Frame Relay PVC bundle member.
pvc (Frame Relay VC-bundle)	Creates a PVC and PVC bundle member and enters Frame Relay VC-bundle-member configuration mode.

efci-bit

To set the explicit forward congestion indication (EFCI) bit field in the ATM cell header for FRF.8 service interworking, use the **efci-bit** command in FRF.8 connect mode. To disable or reset this bit, use the **no** form of this command.

efci-bit {**0**| **map-fecn**}

no efci-bit {**0**| **map-fecn**}

Syntax Description

0	The EFCI field in the ATM cell header is set to 0.
map-fecn	The EFCI field in the ATM cell header is set to 1 when the forward explicit congestion notification (FECN) field in the Frame Relay header is set.

Command Default

The default is **0**.

Command Modes

FRF.8 connect configuration

Command History

Release	Modification
12.1(2)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

This command maps from Frame Relay to ATM.

Examples

The following example creates a connection that connects Frame Relay DLCI 100 to ATM PVC 0/32, and sets the EFCI field in the ATM cell header to 1 when the FECN field in the Frame Relay header is set:

```
Router(config)#
interface atm1/0
Router
(config-if)# pvc 0/32
Router
(config-if)# encapsulation aal5mux fr-atm-srv
Router(config)#
connect serial0 100 atm1/0 0/32 service-interworking
```

```
Router
(config-frf8)# efci-bit map-fecn
```

Related Commands

Command	Description
clp-bit	Sets the ATM CLP field in the ATM cell header.
connect (FRF.8)	Connects a Frame Relay DLCI to an ATM PVC.
connect (FRF.5)	Sets the Frame Relay DE bit field in the Frame Relay cell header.
service translation	Allows mapping between encapsulated ATM PDUs and encapsulated Frame Relay PDUs.

empty-ssl-fragment-insertion

To generate and send an empty Secure Sockets Layer (SSL) fragment to a client as the first encrypted message, use the **empty-ssl-fragment-insertion** command in WAAS SSL configuration mode. To disable this function, use the **no** form of this command.

empty-ssl-fragment-insertion

no empty-ssl-fragment-insertion

Syntax Description

This command has no arguments or keywords.

Command Default

An empty SSL fragment is generated by default and sent to a client as the first encrypted message.

Command Modes

WAAS SSL configuration (config-waas-ssl)

Command History

Release	Modification
15.2(4)M	This command was introduced.

Usage Guidelines

When an SSL connection is optimized by the SSL-Express accelerator, Wide-Area Application Services (WAAS) Express generates an empty SSL fragment and sends it to a client as the first encrypted message. This behavior can impact interoperability with older versions of client applications such as Internet Explorer 6. You can disable the generation and sending of this empty SSL fragment using the **no** form of this command.

Examples

The following example shows how to disable the generation and sending of an empty SSL fragment to a client as the first encrypted message:

```
Device# configure terminal
Device(config)# interface GigabitEthernet0/0
Device(config-if)# waas enable
Device(config-if)# exit
Device(config)# parameter-map type waas waas_global
Device(config-profile)# accelerator ssl-express
Device(config-waas-ssl)# no empty-ssl-fragment-insertion
Device(config-waas-ssl)# end
```

You can use the **show parameter-map type waas** command to verify that the generation of the empty SSL fragment has been disabled.

Related Commands

Command	Description
accelerator ssl-express	Enters WAAS SSL configuration mode and allows the configuration of SSL-Express accelerator parameters.

Command	Description
interface	Configures an interface type and enters interface configuration mode.
parameter-map type waas	Configures WAAS Express global parameters.
show parameter-map type waas	Displays WAAS Express global parameters.
waas enable	Enables WAAS Express on a WAN interface.

encapsulation (Any Transport over MPLS)

To configure the ATM adaptation layer (AAL) encapsulation for an Any Transport over MPLS (AToM), use the **encapsulation** command in the appropriate configuration mode. To remove the ATM encapsulation, use the **no** form of this command.

encapsulation *layer-type*

no encapsulation *layer-type*

Syntax Description

layer-type	The adaptation layer type, which is one of the following: • **aal5** --ATM adaptation layer 5 • **aal0** --ATM adaptation layer 0

Command Default

The default encapsulation is AAL5.

Command Modes

L2transport PVC configuration--for ATM PVCs VC class configuration--for VC class

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.
12.0(30)S	This command was updated to enable ATM encapsulations as part of a virtual circuit (VC) class.
12.0(31)S	This command was integrated into Cisco IOS Release 12.0(31)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.2(33)SRC	This command was integrated into Cisco IOS Release 12.2(33)SRC.
15.0(1)S	This command was integrated into Cisco IOS Release 15.0(1)S.

Release	Modification
Cisco IOS XE Release 3.1S	This command was integrated into Cisco IOS XE Release 3.1S.

Usage Guidelines

In L2transport VC configuration mode, the **pvc** command and the **encapsulation** command work together. Use the commands for AToM differently than for all other applications. The table below shows the differences in how the commands are used.

Table 10: AToM-Specific Variations of the pvc and encapsulation Commands

Other Applications	AToM
`Router(config-if)# pvc 1/100` `Router(config-if-atm-vc)# encapsulation aal5snap`	`Router(config-if)# pvc 1/100 l2transport` `Router(config-if-atm-l2trans-pvc)# encapsulation aal5`

The following list highlights the differences:

- **pvc** command: For most applications, you create a permanent virtual circuit (PVC) by using the **pvc** *vpi/vci* command. For AToM, you must add the **l2transport** keyword to the **pvc** command. The **l2transport** keyword enables the PVC to transport Layer 2 packets.
- **encapsulation** command: The **encapsulation** command for AToM has only two keyword values: **aal5** or **aal0**. You cannot specify an encapsulation type, such as **aal5snap**. In contrast, the **encapsulation aal5** command you use for most other applications requires you to specify the encapsulation type, such as **aal5snap**.
- You cannot create switched virtual circuits or VC bundles to transport Layer 2 packets.

When you use the **aal5** keyword, incoming cells (except Operation, Administration, and Maintenance [OAM] cells) on that PVC are treated as AAL5 encapsulated packets. The router reassembles the packet from the incoming cells. The router does not check the contents of the packet, so it does not need to know the encapsulation type (such as **aal5snap** and **aal5mux**). After imposing the Multiprotocol Label Switching (MPLS) label stack, the router sends the reassembled packet over the MPLS core network.

When you use the **aal0** keyword, the router strips the header error control (HEC) byte from the cell header and adds the MPLS label stack. The router sends the cell over the MPLS core network.

Examples

The following example shows how to configure a PVC to transport ATM cell relay packets for AToM:

```
Router> enable
Router# configure terminal
Router(config)# interface atm1/0
Router(config-if)# pvc 1/100 l2transport
Router(config-if-atm-l2trans-pvc)# encapsulation aal0
Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls
```

The following example shows how to configure ATM AAL5 over MPLS in VC class configuration mode. The VC class is applied to a PVC.

```
Router> enable
Router# configure terminal
```

```
Router(config)# vc-class atm aal5class
Router(config-vc-class)# encapsulation aal5
Router(config)# interface atm1/0
Router(config-if)# pvc ½00 l2transport
Router(config-if-atm-l2trans-pvc)# class-vc aal5class
Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls
```

Related Commands

Command	Description
pvc	Creates or assigns a name to an ATM PVC.

encapsulation (Frame Relay VC-bundle)

To override the encapsulation for a point-to-point subinterface and configure Frame Relay encapsulation for an individual Frame Relay permanent virtual circuit (PVC) bundle, use the **encapsulation**command in Frame Relay VC-bundle configuration mode. To disable the encapsulation for the individual PVC bundle and revert to the encapsulation for the point-to-point subinterface, use the **no** form of this command.

encapsulation [**cisco**| **ietf**]

no encapsulation [**cisco**| **ietf**]

Syntax Description

cisco	(Optional) Uses Cisco proprietary encapsulation, which is a four-byte header, with two bytes to identify the data-link connection identifier (DLCI) and two bytes to identify the packet type
ietf	(Optional) Sets the encapsulation method to comply with the Internet Engineering Task Force (IETF) standard (RFC 1490 and RFC 2427). Use this keyword when connecting to another vendor's equipment across a Frame Relay network on point-to-point interfaces.

Command Default

Encapsulation type that is configured on the main interface.

Command Modes

Frame Relay VC-bundle configuration

Command History

Release	Modification
12.2(13)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.

Usage Guidelines

Use this command to override the encapsulation at a point-to-point subinterface for an individual Frame Relay PVC bundle. This command is available for point-to-point subinterfaces only; it cannot be used on multipoint interfaces.

Examples

The following example configures RFC 1490 encapsulation for the Frame Relay PVC bundle named "P2P-5":

```
interface serial 1/4.2 point-to-point
 ip address 10.1.1.1 255.0.0.0
```

```
frame-relay vc-bundle P2P-5
 encapsulation ietf
```

Related Commands

Command	Description
encapsulation frame-relay	Enables Frame Relay encapsulation on an interface.

encapsulation (L2TP)

To specify the Layer 2 data encapsulation method to be used for tunneling IP traffic over a pseudowire, use the **encapsulation**(L2TP) command in pseudowire class configuration mode. To remove the specified Layer 2 encapsulation method, use the **no** form of this command.

encapsulation {**l2tpv2**| **l2tpv3 [manual]**| **mpls**}

no encapsulation {**l2tpv2**| **l2tpv3 [manual]**| **mpls**}

Syntax Description

l2tpv2	Uses Layer 2 Tunneling Protocol (L2TP) as the tunneling method to encapsulate data in the pseudowire.
l2tpv3	Uses Layer 2 Tunneling Protocol Version 3 (L2TPv3) as the tunneling method to encapsulate data in the pseudowire.
manual	(Optional) No signaling is to be used in the L2TPv3 control channel.
mpls	Uses Multiprotocol Label Switching (MPLS) as the tunneling method to encapsulate data in the pseudowire.

Command Default

No encapsulation method is specified.

Command Modes

Pseudowire class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.3(2)T	The **l2tpv2**keyword was added and this command was integrated into Cisco IOS Release 12.3(2)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

This command must be configured if the pseudowire class will be referenced from an xconnect or pseudowire configured to forward Layer 2 traffic.

Examples

The following example shows how to configure L2TPv3 as the data encapsulation method for the pseudowire class named “ether-pw”:

```
Router(config)
# pseudowire-class ether-pw
Router(config-pw)
# encapsulation l2tpv3
```

Related Commands

Command	Description
pseudowire-class	Specifies the name of an L2TP pseudowire class and enters pseudowire class configuration mode.

encapsulation (Layer 2 local switching)

To configure the ATM adaptation layer (AAL) for a Layer 2 local switching ATM permanent virtual circuit (PVC), use the **encapsulation** command in ATM PVC L2transport configuration mode. To remove an encapsulation from a PVC, use the **no** form of this command.

encapsulation *layer-type*

no encapsulation *layer-type*

Syntax Description

layer-type	Adaptation layer type. The values are: • **aal5** • **aal0** • **aal5snap** • **aal5mux** • **aal5nlpid** (not available on Cisco 12000 series)

Command Default

If you do not create a PVC, one is created for you. The default encapsulation types for autoprovisioned PVCs are as follows:

- For ATM-to-ATM local switching, the default encapsulation type for the PVC is AAL0.
- For ATM-to-Ethernet or ATM-to-Frame Relay local switching, the default encapsulation type for the PVC is AAL5 SNAP.

Command Modes

ATM PVC L2transport configuration

Command History

Release	Modification
12.0(27)S	This command was introduced for Layer 2 local switching.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.0(30)S	This command was integrated into Cisco IOS Release 12.0(30)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Release	Modification
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

The pvc command and the encapsulation command work together. The use of these commands with Layer 2 local switching is slightly different from the use of these commands with other applications. The following list highlights the differences:

- For Layer 2 local switching, you must add the **l2transport** keyword to the **pvc** command. The **l2transport** keyword enables the PVC to transport Layer 2 packets.
- The Layer 2 local switching **encapsulation** command works only with the **pvc** command. You cannot create switched virtual circuits or VC bundles to transport Layer 2 packets. You can use only PVCs to transport Layer 2 packets.

The table below shows the encapsulation types supported for each transport type:

Table 11: Supported Encapsulation Types

Interworking Type	Encapsulation Type
ATM to ATM	AAL0, AAL5
ATM to Ethernet with IP interworking	AAL5SNAP, AAL5MUX
ATM to Ethernet with Ethernet interworking	AAL5SNAP
ATM to Frame-Relay	AAL5SNAP, AAL5NLPID

Examples

The following example shows how to configure a PVC to transport AAL0 packets for Layer 2 local switching:

```
pvc 1/100 l2transport
 encapsulation aal0
```

Related Commands

Command	Description
pvc	Creates or assigns a name to an ATM PVC.

encapsulation default

To configure the default service instance on a port, use the **encapsulation default** command in service instance mode. To delete the default service instance on a port, use the **no** form of this command.

encapsulation default

no encapsulation default

Syntax Description

This command has no arguments or keywords.

Command Default

No default service instance is configured on the port.

Command Modes

Service instance

Command History

Release	Modification
12.2(33)SRB	This command was introduced.
15.3(2)S	This command was implemented on the Cisco ASR 901 Series Aggregation Services Routers.

Usage Guidelines

If the default service instance is the only one configured on a port, the encapsulation default command matches all ingress frames on that port. If the default service instance is configured on a port that has other non-default service instances, the encapsulation default command matches frames that are unmatched by those non-default service instances (anything that does not meet the criteria of other services instances on the same physical interface falls into this service instance).

Only a single default service instance can be configured per interface. If you attempt to configure more than one default service instance per interface, the encapsulation default command is rejected.

Only one encapsulation command must be configured per service instance.

Examples

The following example shows how to configure a service instance on a port:

```
Device(config-if-srv)# encapsulation default
```

Related Commands

Command	Description
encapsulation dot1q (service instance)	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance.

Command	Description
encapsulation dot1q second-dot1q	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
encapsulation untagged	Defines the matching criteria to map untagged ingress Ethernet frames on an interface to the appropriate service instance.

encapsulation dot1q (service instance)

To define the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance, use the **encapsulation dot1q** command in Ethernet service instance configuration mode. To delete the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance, use the **no** form of this command.

encapsulation dot1q *vlan-id*[**,** *vlan-id*[**-***vlan-id*]][**native**]

no encapsulation dot1q *vlan-id*[**,** *vlan-id*[**-***vlan-id*]][**native**]

Syntax Description

vlan-id	VLAN ID, integer in the range 1 to 4094. A hyphen must be entered to separate the starting and ending VLAN ID values that are used to define a range of VLAN IDs. (Optional) A comma must be entered to separate each VLAN ID range from the next range.
native	(Optional) Sets the VLAN ID value of the port to the value specified by the*vlan-id* argument.

Command Default

No matching criteria are defined.

Command Modes

Ethernet service instance (config-if-srv)

Command History

Release	Modification
12.2(33)SRB	This command was introduced.
Cisco IOS XE Release 3.5S	This command was integrated into Cisco IOS XE Release 3.5S. Support was added for the Cisco ASR 903 Router.

Usage Guidelines

The criteria for this command are: a single VLAN, a range of VLANs, and lists of the previous two.

A single 802.1Q service instance allows one VLAN, multiple VLANs, or a range of VLANs. The native keyword can be set only if a single VLAN tag has been specified.

Only a single service instance per port is allowed to have the **native** keyword.

Only one **encapsulation** command may be configured per service instance.

Examples

The following example shows how to map 802.1Q frames ingress on an interface to the appropriate service instance:

```
Router(config-if-srv)# encapsulation dot1q 10
```

Related Commands

Command	Description
encapsulation default	Configures the default service instance on a port.
encapsulation dot1q second-dot1q	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
encapsulation untagged	Defines the matching criteria to map untagged ingress Ethernet frames on an interface to the appropriate service instance.

encapsulation dot1q second-dot1q

To define the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance, use the **encapsulation dot1q second-dot1q** command in service instance mode. To delete the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance, use the **no** form of this command.

encapsulation dot1q *vlan-id* **second-dot1q**[**any**| *vlan-id*[**,** *vlan-id*[*-vlan-id*]]]

no encapsulation dot1q *vlan-id* **second-dot1q**[**any**| *vlan-id*[**,** *vlan-id*[*-vlan-id*]]]

Syntax Description

vlan-id	VLAN ID, integer in the range 1 to 4094. Hyphen must be entered to separate the starting and ending VLAN ID values that are used to define a range of VLAN IDs. (Optional) Comma must be entered to separate each VLAN ID range from the next range.
any	Any second tag in the range 1 to 4094.

Command Default

No matching criteria are defined.

Command Modes

Service instance (config-if-srv)

Command History

Release	Modification
12.2(33)SRB	This command was introduced.
15.1(2)SNH	This command was integrated into Cisco IOS Release 15.1(2)SNH to provide support for Cisco ASR 901 Series Aggregation Services Routers.

Usage Guidelines

The criteria for this command are: the outer tag must be unique and the inner tag may be a single VLAN, a range of VLANs or lists of the previous two.

QinQ service instance, allows single, multiple or range on second-dot1q.

Only one encapsulation command must be configured per service instance.

Examples

The following example shows how to map ingress frames to a service instance:

```
Device(config-if-srv)# encapsulation dot1q second-dot1q 20
```

Related Commands

Command	Description
encapsulation default	Configures the default service instance on a port.
encapsulation dot1q (service instance)	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance.
encapsulation untagged	Defines the matching criteria to map untagged ingress Ethernet frames on an interface to the appropriate service instance.

encapsulation frame-relay

To enable Frame Relay encapsulation, use the **encapsulation frame-relay** command in interface configuration mode. To disable Frame Relay encapsulation, use the **no** form of this command.

encapsulation frame-relay [**cisco**| **ietf**]

no encapsulation frame-relay [ietf]

Syntax Description

cisco	(Optional) Uses Cisco's own encapsulation, which is a 4-byte header, with 2 bytes to identify the data-link connection identifier (DLCI) and 2 bytes to identify the packet type.
ietf	(Optional) Sets the encapsulation method t o comply with the Internet Engineering Task Force (IETF) standard (RFC 1490). Use this keyword when connecting to another vendor's equipment across a Frame Relay network.

Command Default

The default is Cisco's own encapsulation.

Command Modes

Interface configuration

Command History

Release	Modification
10.0	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use this command with no keywords to restore the default Cisco encapsulation, which is a 4-byte header with 2 bytes for the DLCI and 2 bytes to identify the packet type.

You should shut down the interface prior to changing encapsulation types. Although this is not required, shutting down the interface ensures that the interface is reset for the new encapsulation.

Examples

The following example configures Cisco Frame Relay encapsulation on interface serial 1:

```
interface serial 1
 encapsulation frame-relay
```

Use the **ietf** keyword if your router or access server is connected to another vendor's equipment across a Frame Relay network to conform with RFC 1490:

```
interface serial 1
 encapsulation frame-relay ietf
```

encapsulation frame-relay mfr

To create a multilink Frame Relay bundle link and to associate the link with a bundle, use the **encapsulation frame-relay mfr**command in interface configuration mode. To remove the bundle link from the bundle, use the **no** form of this command.

encapsulation frame-relay mfr *number* [*name*]

no encapsulation frame-relay mfr *number* [*name*]

Syntax Description

number	Interface number of the multilink Frame Relay bundle with which this bundle link will be associated.
name	(Optional) Bundle link identification (LID) name. The name can be up to 49 characters long. The default is the name of the physical interface.

Command Default

Frame Relay encapsulation is not enabled.

Command Modes

Interface configuration

Command History

Release	Modification
12.0(17)S	This command was introduced on the Cisco 12000 series routers.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.0(24)S	This command was implemented on VIP-enabled Cisco 7500 series routers.
12.3(4)T	Support for this command on VIP-enabled Cisco 7500 series routers was integrated into Cisco IOS Release 12.3(4)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.0(33)S	Support for IPv6 was added. This command was implemented on the Cisco 12000 series routers.

Usage Guidelines

Use the *name* argument to assign a LID name to a bundle link. This name will be used to identify the bundle link to peer devices and to enable the devices to determine which bundle links are associated with which bundles. The LID name can also be assigned or changed by using the **frame-relay multilink lid** command on the bundle link interface. If the LID name is not assigned, the default name is the name of the physical interface.

Tip To minimize latency that results from the arrival order of packets, we recommend bundling physical links of the same line speed in one bundle.

To remove a bundle link from a bundle, use the **no encapsulation frame-relay mfr** command or configure a new type of encapsulation on the interface by using the **encapsulation** command.

Examples

The following example shows serial interface 0 being associated as a bundle link with bundle interface “mfr0.” The bundle link identification name is “BL1.”

```
interface mfr0
!
interface serial 0
 encapsulation frame-relay mfr0 BL1
```

Related Commands

Command	Description
debug frame-relay multilink	Displays debug messages for multilink Frame Relay bundles and bundle links.
encapsulation	Sets the encapsulation method used by the interface.
frame-relay multilink lid	Assigns a LID name to a multilink Frame Relay bundle link.
show frame-relay multilink	Displays configuration information and statistics about multilink Frame Relay bundles and bundle links.

encapsulation l2tpv3

To specify that Layer 2 Tunnel Protocol Version 3 (L2TPv3) is used as the data encapsulation method for tunneling IP traffic over the pseudowire, use the **encapsulation l2tpv3** command in pseudowire class or VC class configuration mode. To remove L2TPv3 as the encapsulation method, use the **no pseudowire-class** command (see the Usage Guidelines for more information).

encapsulation l2tpv3

no pseudowire-class

Syntax Description

This command has no arguments or keywords.

Command Default

No encapsulation method is specified.

Command Modes

Pseudowire class configuration VC class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.3(2)T	This command was integrated into Cisco IOS Release 12.3(2)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(27)SBC	Support for this command was integrated into Cisco IOS Release 12.2(27)SBC.

Usage Guidelines

This command must be configured if the pseudowire class will be referenced from an Xconnect configured to forward L2TPv3 traffic.

Once you specify the **encapsulation l2tpv3**command, you cannot remove it using the **no encapsulation l2tpv3** command. Nor can you change the command's setting using the **encapsulation mpls** command. Those methods result in the following error message:

```
Encapsulation changes are not allowed on an existing pw-class.
```

To remove the command, you must delete the pseudowire with the **no pseudowire-class** command. To change the type of encapsulation, remove the pseudowire with the **no pseudowire-class** command and re-establish the pseudowire and specify the new encapsulation type.

Examples

The following example shows how to configure L2TPv3 as the data encapsulation method for the pseudowire class named ether-pw:

```
Router(config)
```

```
# pseudowire-class ether-pw
Router(config-pw)
# encapsulation l2tpv3
```

The following example configures ATM AAL5 over L2TPv3 in VC class configuration mode:

```
vc-class atm aal5class
 encapsulation aal5
```

Related Commands

Command	Description
encapsulation mpls	Configures MPLS as the data encapsulation method over AToM-enabled IP/MPLS networks.
pseudowire-class	Specifies the name of an L2TP pseudowire class and enters pseudowire class configuration mode.

encapsulation lapb

To exchange datagrams over a serial interface using Link Access Procedure, Balanced (LAPB) encapsulation, use the **encapsulation lapb**command in interface configuration mode.

encapsulation lapb [**dte**| **dce**] [**multi**| *protocol*]

Syntax Description

dte	(Optional) Specifies operation as a data terminal equipment (DTE) device. This is the default LAPB mode.
dce	(Optional) Specifies operation as a data communications equipment (DCE) device.
multi	(Optional) Specifies use of multiple LAN protocols to be carried on the LAPB line.
protocol	(Optional) A single protocol to be carried on the LAPB line. A single protocol can be one of the following: **appletalk**, **clns** (ISO CLNS), **decnet**, **ip**, and **ipx** (Novell IPX).IP is the default protocol.

Command Default

The default serial encapsulation is High-Level Data Link Control (HDLC). You must explicitly configure a LAPB encapsulation method.

DTE operation is the default LAPB mode. IP is the default protocol.

Command Modes

Interface configuration

Command History

Release	Modification
10.0	This command was introduced.
10.3	The following keywords and argument were introduced: **dte, dce, multi,** *protocol*.
12.2(13)T	The **apollo**, **vines**, and **xns** arguments were removed because support for Apollo Domain, Banyan VINES, and Xerox Network Systems is no longer available in the Cisco IOS software.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

LAPB encapsulations are appropriate only for private connections, where you have complete control over both ends of the link. Connections to X.25 networks should use an X.25 encapsulation configuration, which operates the X.25 Layer 3 protocol above a LAPB Layer 2.

One end of the link must be a logical DCE device, and the other end a logical DTE device. (This assignment is independent of the interface's hardware DTE or DCE identity.)

Both ends of the LAPB link must specify the same protocol encapsulation.

LAPB encapsulation is supported on serial lines configured for dial-on-demand routing (DDR). It can be configured on DDR synchronous serial and ISDN interfaces and on DDR dialer rotary groups. It is not supported on asynchronous dialer interfaces.

A single-protocol LAPB encapsulation exchanges datagrams of the given protocol, each in a separate LAPB information frame. You must configure the interface with the protocol-specific parameters needed--for example, a link that carries IP traffic will have an IP address defined for the interface.

A multiprotocol LAPB encapsulation can exchange any or all of the protocols allowed for a LAPB interface. It exchanges datagrams, each in a separate LAPB information frame. Two bytes of protocol identification data precede the protocol data. You need to configure the interface with all the protocol-specific parameters needed for each protocol carried.

Multiprotocol LAPB encapsulation supports transparent bridging. This feature requires use of the **encapsulation lapb multi**command followed by the **bridge-group** command, which identifies the bridge group associated with multiprotocol LAPB encapsulation. This feature does *not* support use of the **encapsulation lapb** *protocol* command with a **bridge** keyword.

LAPB encapsulation supports the priority and custom queueing features.

Examples

The following example sets the operating mode as DTE and specifies that AppleTalk protocol traffic will be carried on the LAPB line:

```
interface serial 1
 encapsulation lapb dte appletalk
```

Related Commands

Command	Description
bridge-group	Assigns each network interface to a bridge group.

encapsulation smds

To enable Switched Multimegabit Data Service (SMDS) on the desired interface, use the **encapsulation smds**interface configuration command.

encapsulation smds

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled

Command Modes

Interface configuration

Command History

Release	Modification
10.0	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

The interface to which this command applies must be a serial interface. All subsequent SMDS configuration commands apply only to an interface with encapsulation SMDS.

Note The maximum packet size allowed in the SMDS specifications (TA-772) is 9188. This is larger than the packet size used by servers with most media. The Cisco default maximum transmission unit (MTU) size is 1500 bytes to be consistent with Ethernet. However, on the High Speed Serial Interface (HSSI), the default MTU size is 4470 bytes. If a larger MTU is used, the **mtu**command must be entered before the **encapsulation smds** command.

Caution The Cisco MCI card has buffer limitations that prevent setting the MTU size higher than 2048, and the HSSI card has buffer limitations that prevent setting the MTU size higher than 4500. Configuring higher settings can cause inconsistencies and performance problems.

Examples

The following example shows how to configure the SMDS service on serial interface 0:

```
interface serial 0
 encapsulation smds
```

Related Commands

Command	Description
mtu	Adjusts the maximum packet size or MTU size.

encapsulation untagged

To define the matching criteria to map untagged ingress Ethernet frames on an interface to the appropriate service instance, use the **encapsulation untagged**command in the service instance mode. To delete the matching criteria to map untagged ingress Ethernet frames on an interface to the appropriate service instance, use the **no** form of this command.

encapsulation untagged

no encapsulation untagged

Syntax Description

This command has no arguments or keywords.

Command Default

No matching criteria are defined.

Command Modes

Service instance (config-if-srv)

Command History

Release	Modification
12.2(33)SRB	This command was introduced.
15.1(2)SNG	This command was implemented on Cisco ASR 901 Series Aggregation Services Routers.
15.2(02)SA	This command was implemented on the Cisco ME 2600X Series Ethernet Access Switches.

Usage Guidelines

Only one service instance per port is allowed to have untagged encapsulation. The reason is to be able to unambiguously map the incoming frames to the service instance. However, it is possible for a port that hosts a service instance matching untagged traffic to host other service instances that match tagged frames.

Only one encapsulation command may be configured per service instance.

Examples

The following example shows how to map untagged ingress Ethernet frames to a service instance:

```
Device(config-if-srv)# encapsulation untagged
```

Related Commands

Command	Description
encapsulation default	Configures the default service instance on a port.

Command	Description
encapsulation dot1ad	Defines the matching criteria to be used in order to map single-tagged 802.1ad frames ingress on an interface to the appropriate service instance. The criteria for this command are single VLAN, range of VLANs, and lists of these two.
encapsulation dot1q (service instance)	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance.
encapsulation dot1q second-dot1q	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.

encapsulation x25

To specify a serial interface's operation as an X.25 device, use the **encapsulation x25** command in interface configuration mode. To remove the specification, use the **no** form of this command.

encapsulation x25 [**dte**| **dce**] [**ddn**| **bfe**| **ietf**]

no encapsulation x25 [**dte**| **dce**] [**ddn**| **bfe**| **ietf**]

Syntax Description

dte	(Optional) Specifies operation as a data terminal equipment (DTE). This is the default X.25 mode.
dce	(Optional) Specifies operation as a data communications equipment (DCE).
ddn	(Optional) Specifies Defense Data Network (DDN) encapsulation on an interface using DDN X.25 Standard Service.
bfe	(Optional) Specifies Blacker Front End (BFE) encapsulation on an interface attached to a BFE device.
ietf	(Optional) Specifies that the interface's datagram encapsulation defaults to use of the Internet Engineering Task Force (IETF) standard method, as defined by RFC 1356.

Command Default

The default serial encapsulation is High-Level Data Link Control (HDLC). You must explicitly configure an X.25 encapsulation method.

DTE operation is the default X.25 mode. Cisco's traditional X.25 encapsulation method is the default.

Command Modes

Interface configuration

Command History

Release	Modification
10.0	This command was introduced.

Release	Modification
10.3	The following keywords were added: • **dte** • **dce** • **ddn** • **bfe** • **ietf**
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

One end of an X.25 link must be a logical DCE device and the other end a logical DTE device. (This assignment is independent of the interface's hardware DTE or DCE identity.) Typically, when connecting to a public data network (PDN), the customer equipment acts as the DTE device and the PDN attachment acts as the DCE.

Cisco has long supported the encapsulation of a number of datagram protocols, using a standard means when available and a proprietary means when necessary. The IETF adopted a standard, RFC 1356, for encapsulating most types of datagram traffic over X.25. X.25 interfaces use Cisco's traditional method unless explicitly configured for IETF operation; if the **ietf** keyword is specified, that standard is used unless Cisco's traditional method is explicitly configured. For details see the **x25 map** command.

You can configure a router attaching to the DDN or to a BFE device to use their respective algorithms to convert between IP and X.121 addresses by using the **ddn** or **bfe** option, respectively. An IP address must be assigned to the interface, from which the algorithm will generate the interface's X.121 address. For proper operation, this X.121 address must not be modified.

A router DDN attachment can operate as either a DTE or a DCE device. A BFE attachment can operate only as a DTE device. The **ietf** option is not available if either the **ddn** or **bfe** option is selected.

Examples

The following example configures the interface for connection to a BFE device:

```
interface serial 0
 encapsulation x25 bfe
```

Related Commands

Command	Description
x25 map	Sets up the LAN protocols-to-remote host mapping.

ethernet evc

To define an Ethernet virtual connection (EVC) and to enter EVC configuration mode, use the **ethernet evc** command in global configuration mode. To delete the EVC, use the **no** form of this command.

ethernet evc *evc-id*

no ethernet evc *evc-id*

Syntax Description

evc-id	String from 1 to 100 characters that identifies the EVC.

Command Default

No EVCs are defined.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.2(25)SEG	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.4(15)T	This command was integrated into Cisco IOS Release 12.4(15)T.
Cisco IOS XE Release 3.8S	This command was integrated into Cisco IOS XE Release 3.8S.
15.1(2)SNG	This command was implemented on the Cisco ASR 901 Series Aggregation Services Router.

Usage Guidelines

After you enter the **ethernet evc** command, the device enters EVC configuration mode and the following configuration commands are available:

- **default** --S ets the EVC to its default states.
- **exit** -- Exits EVC configuration mode and returns the CLI to global configuration mode.
- **no** -- Negates a command or returns a command to its default setting.
- **oam protocol** -- Configures the Ethernet operations, administration, and maintenance (OAM) protocol and sets parameters.
- **uni count** -- Configures a UNI count for the EVC.

Examples

The following example shows how to define an EVC named test1 and to enter EVC configuration mode:

```
Device(config)# ethernet evc test1
Device(config-evc)#
```

Related Commands

Command	Description
oam protocol	Configures the EVC OAM protocol.
service instance	Configures an Ethernet service instance and attaches an EVC to it.
show ethernet service evc	Displays information about configured EVCs.
uni count	Sets the UNI count for an EVC.

exp

To configure Multiprotocol Label Switching (MPLS) experimental (EXP) levels for a Frame Relay permanent virtual circuit (PVC) bundle member, use the **exp** command in Frame Relay VC-bundle-member configuration mode. To remove the EXP level configuration from the PVC, use the **no** form of this command.

exp {*level*| **other**}

no exp

Syntax Description

level	The MPLS EXP level or levels for this Frame Relay PVC bundle member. The range is from 0 to 7. A PVC bundle member can be configured with a single level, multiple individual levels, a range of levels, multiple ranges of levels, or a combination of individual levels and level ranges. Levels can be specified in ascending or descending order (although a subsequent **show running-config**command will display them in ascending order). Examples are as follows: • 0 • 0,2,3 • 6-5 • 0-2,4-5 • 0,1,2-4,7
other	Specifies that this Frame Relay PVC bundle member will handle all of the remaining MPLS EXP levels that are not explicitly configured on any other bundle member PVCs.

Command Default

EXP levels are not configured.

Command Modes

Frame Relay VC-bundle-member configuration

Command History

Release	Modification
12.2(13)T	This command was introduced.

Release	Modification
12.2(16)BX	This command was integrated into Cisco IOS Release 12.2(16)BX.
12.0(26)S	This command was integrated into Cisco IOS Release 12.0(26)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.

Usage Guidelines

Assignment of MPLS EXP levels to Frame Relay PVC bundle members lets you create differentiated services, because you can distribute the levels over the various PVC bundle members. You can map a single level or a range of levels to each discrete PVC in the bundle, which enables PVCs in the bundle to carry packets marked with different levels.

Use the **exp other** command to indicate that a PVC can carry traffic marked with EXP levels not specifically configured for other PVCs. Only one PVC in the bundle can be configured using the **exp other** command.

All EXP levels must be accounted for in the PVC bundle configuration, or the bundle will not come up. However, a PVC can be a bundle member but have no EXP level associated with it. As long as all valid EXP levels are handled by other PVCs in the bundle, the bundle can come up, but the PVC that has no EXP level configured will not participate in it.

The **exp** command is available only when MPLS is configured on the interface with the **mpls ip** command.

You can overwrite the EXP level configuration on a PVC by reentering the **exp** command with a new value.

The MPLS experimental bits are a bit-by-bit copy of the IP precedence bits. When Frame Relay PVC bundles are configured for IP precedence and MPLS is enabled, the **precedence** command is replaced by the **exp** command. When MPLS is disabled, the **exp** command is replaced by the **precedence** command.

Examples

The following example shows the configuration of four Frame Relay PVC bundle members in PVC bundle bundle1 configured with MPLS EXP level support:

```
interface serial 0.1 point-to-point
 encapsulation frame-relay
 ip address 10.1.1.1
 mpls ip
 frame-relay vc-bundle bundle1
 pvc 100 ny-control
 class control
 exp 7
 protect vc
 pvc 101 ny-premium
 class premium
 exp 6-5
 protect group
 no bump traffic
 bump explicit 7
 pvc 102 my-priority
 class priority
 exp 4-2
 protect group
 pvc 103 ny-basic
 class basic
 exp other
 protect group
```

Related Commands

Command	Description
bump	Configures the bumping rules for a specific PVC member of a bundle.
class	Associates a map class with a specified DLCI.
dscp (Frame Relay VC-bundle-member)	Configures the DSCP value or values for a Frame Relay PVC bundle member.
match	Specifies which bits of the IP header to use for mapping packet service levels to Frame Relay PVC bundle members.
mpls ip	Enables label switching of IPv4 packets on an interface.
precedence (Frame Relay VC-bundle-member)	Configures the precedence levels for a Frame Relay PVC bundle member.
protect	Configures a Frame Relay PVC bundle member with protected group or protected PVC status.