

## debug tag-switching adjacency

The **debug tag-switching adjacency** command is replaced by the **debug mpls adjacency** command. See the [debug mpls adjacency](#) command for more information.

# debug tag-switching atm-cos

To display ATM label-VC bind or request activity based on the configuration of a CoS map, use the **debug tag-switching atm-cos** ATM privileged EXEC command.

**debug tag-switching atm-cos** [*bind* | *request*]

Syntax Description		
	<i>bind</i>	Specifies debug information about bind responses for a VC path.
	<i>request</i>	Specifies debug information about bind requests for a VC path.

Command History	Release	Modification
	12.0(5)T	This command was introduced.

## Examples

The following is sample output from the **debug tag-switching atm-cos** command.

```
Router# show tag forwarding

Local Outgoing Prefix      Bytes tag Outgoing Next Hop
tag tag or VC or Tunnel Id switched interface
26 28 17.17.17.17/32 0 PO6/0 point2point
27 Pop tag 11.11.11.11/32 1560 PO6/0 point2point
28 27 16.16.16.16/32 0 PO6/0 point2point
29 30 92.0.0.0/8 0 PO6/0 point2point
30 Pop tag 95.0.0.0/8 2600 PO6/0 point2point
31 2/34 10.10.10.10/32 0 AT2/0.1 point2point
32 Pop tag 14.14.14.14/32 0 Fa5/0 91.0.0.1
33 Pop tag 90.0.0.0/8 0 Fa5/0 91.0.0.1
34 Pop tag 96.0.0.0/8 0 Fa5/0 91.0.0.1
    2/36 96.0.0.0/8 0 AT2/0.1 point2point
35 35 93.0.0.0/8 0 PO6/0 point2point
36 36 12.12.12.12/32 0 PO6/0 point2point
37 37 15.15.15.15/32 0 PO6/0 point2point
38 37 18.18.18.18/32 0 Fa5/0 91.0.0.1
39 39 97.0.0.0/8 540 PO6/0 point2point
40 40 98.0.0.0/8 0 PO6/0 point2point

Router# debug tag atm-c
Router# debug tag atm-cos ?
  bind Bind response for VC path
  request Requests for VC binds path

Router# debug tag atm-cos bind
ATM TAGCOS Bind response debugging is on

Router# debug tag atm-cos request
ATM TAGCOS VC requests debugging is on

Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface a2/0.1
Router(config-subif)# tag atm multi
Router(config-subif)# end
Router#
19:59:14:%SYS-5-CONFIG_I:Configured from console by console
```

```

Router#
19:59:24:TAGCOS-REQ:vc request 10.10.10.10/32, available
19:59:24:TAGCOS-REQ:vc request 10.10.10.10/32, standard
19:59:24:TAGCOS-REQ:vc request 10.10.10.10/32, premium
19:59:24:TAGCOS-REQ:vc request 10.10.10.10/32, control
19:59:24:TAGCOS-REQ:vc request 96.0.0.0/8, available
19:59:24:TAGCOS-REQ:vc request 96.0.0.0/8, standard
19:59:24:TAGCOS-REQ:vc request 96.0.0.0/8, premium
19:59:24:TAGCOS-REQ:vc request 96.0.0.0/8, control
TAGCOS-REQ/TCATM:11.11.11.11/32, len=4352, band=1099528405504, class=0x700
TAGCOS-REQ/TCATM:12.12.12.12/32, len=4352, band=2199040033280, class=0x700
TAGCOS-REQ/TCATM:13.13.13.13/32, len=4352, band=3298551661056, class=0x700
TAGCOS-REQ/TCATM:14.14.14.14/32, len=4352, band=4398063288832, class=0x700
TAGCOS-REQ/TCATM:15.15.15.15/32, len=4352, band=5497574916608, class=0x700
TAGCOS-REQ/TCATM:16.16.16.16/32, len=4352, band=6597086544384, class=0x700
TAGCOS-REQ/TCATM:17.17.17.17/32, len=4352, band=7696598172160, class=0x700
TAGCOS-REQ/TCATM:18.18.18.18/32, len=4352, band=8796109799936, class=0x700
TAGCOS-REQ/TCATM:90.0.0.0/8, len=768, band=3940649674539009, class=0x2
TAGCOS-REQ/TCATM:91.0.0.0/8, len=768, band=3940649674604545, class=0x2
TAGCOS-REQ/TCATM:92.0.0.0/8, len=768, band=3940649674670081, class=0x2
TAGCOS-REQ/TCATM:93.0.0.0/8, len=768, band=3940649674735617, class=0x2
TAGCOS-REQ/TCATM:94.0.0.0/8, len=768, band=3940649674801153, class=0x2
TAGCOS-REQ/TCATM:95.0.0.0/8, len=768, band=3940649674866689, class=0x2
TAGCOS-REQ/TCATM:97.0.0.0/8, len=768, band=3940649674932225, class=0x2
TAGCOS-REQ/TCATM:98.0.0.0/8, len=768, band=3940649674997761, class=0x2
TAGCOS-BIND:binding_ok 10.10.10.10/32, VCD=41 - control 41,41,41,41
TAGCOS-BIND:binding_ok 10.10.10.10/32, Inform TFIB pidx=0, in_tag=31, idx=0x80000000
TAGCOS-BIND:binding_ok 96.0.0.0/8, VCD=42 - control 42,42,42,42
TAGCOS-BIND:binding_ok 96.0.0.0/8, Inform TFIB pidx=1, in_tag=34, idx=0x80000001
TAGCOS-BIND:binding_ok 10.10.10.10/32, VCD=43 - premium 43,43,43,41
TAGCOS-BIND:binding_ok 96.0.0.0/8, VCD=44 - premium 44,44,44,42
TAGCOS-BIND:binding_ok 10.10.10.10/32, VCD=45 - standard 45,45,43,41
TAGCOS-BIND:binding_ok 96.0.0.0/8, VCD=46 - standard 46,46,44,42
TAGCOS-BIND:binding_ok 10.10.10.10/32, VCD=47 - available 47,45,43,41
TAGCOS-BIND:binding_ok 96.0.0.0/8, VCD=48 - available 48,46,44,42
72k-41-5#
72k-41-5#

```

**Related Commands**

Command	Description
<b>debug tag atm-tdp</b>	Debugs label-controlled ATM TDP.
<b>debug tag packets</b>	Debugs tag switching packets.
<b>debug tag tdp</b>	Debugs tag distribution protocol items and information.

## debug tag-switching atm-tdp api

To display information about the VCI allocation of tag VCs (TVCs), free, and cross-connect requests, use the **debug tag-switching atm-tdp api** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching atm-tdp api**

**no debug tag-switching atm-tdp api**

### Syntax Description

This command has no arguments or keywords.

### Usage Guidelines

You can use the **debug tag-switching atm-tdp api** command with the **debug tag-switching atm-tdp states** command to display more complete information about a TVC.

### Examples

The following is sample output from the **debug tag-switching atm-tdp api** command:

```
Router# debug tag-switching atm-tdp api

Tailend Router Free tag Req 167.50.0.0 on ATM0/0.2 VPI/VCI 1/674
TAGATM_API: received tag free request
    interface: ATM0/0.2 dir: in vpi: 1 vci: 674
TAGATM_API: completed tag free
    interface: ATM0/0.2 vpi: 1 vci: 674
    result: TAGATM_OK
```

[Table 186](#) describes the significant fields shown in the display.

**Table 186** *debug tag-switching atm-tdp api Field Descriptions*

Field	Description
TAGATM_API	Subsystem that prints the message.
interface	Interface used by the driver to allocate or free VPI/VCI resources.
dir	Direction of the VC: <ul style="list-style-type: none"> <li>In—Input or receive VC</li> <li>Out—Output VC</li> </ul>
vpi	Virtual path identifier.
vci	Virtual channel identifier.
result	Return error code from the driver API.

### Related Commands

Command	Description
<a href="#">debug tag-switching atm-tdp states</a>	Displays information about TVC state transitions as they occur.

## debug tag-switching atm-tdp routes

To display information about the state of the routes for which VCI requests are being made, use the **debug tag-switching atm-tdp routes** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching atm-tdp routes**

**no debug tag-switching atm-tdp routes**

### Syntax Description

This command has no arguments or keywords.

### Usage Guidelines

When there are many routes and system activities (that is, shutting down interfaces, learning of new routes, and so on), the **debug tag-switching atm-tdp routes** command displays a substantial amount of information that may interfere with system timing. Most commonly, this affects the normal operation of the Tag Distribution Protocol (TDP). You should increase the TDP hold-time value by using the **tag-switching tdp holdtime** command.

### Examples

The following is sample output from the **debug tag-switching atm-tdp routes** command:

```
Router# debug tag-switching atm-tdp routes

CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.7.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.7.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.7.0.0/16
CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.8.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.8.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.8.0.0/16
CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.9.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.9.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.9.0.0/16
CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.10.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.10.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.10.0.0/16
CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.11.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.11.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.11.0.0/16
```

Table 187 describes the significant fields shown in the display.

**Table 187** *debug tag-switching atm-tdp routes Field Descriptions*

Field	Description
CleanupRoutes	Cleans up the routing table after a route has been deleted.
not deleting route of idb ATM0/0.2	Route cleanup event has not removed the specified route.
rdbIndex	Index identifying the route.
tcatmFindRouteTags	Request a VC for the route.
idb	Internal descriptor for an interface.
nh	Next hop for the route.
index	Identifier for the route.
AddNewRoute	Action of adding routes for a prefix or address.

## debug tag-switching atm-tdp states

To display information about TVC state transitions as they occur, use the **debug tag-switching atm-tdp states** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching atm-tdp states**

**no debug tag-switching atm-tdp states**

### Syntax Description

This command has no arguments or keywords.

### Usage Guidelines

When there are many routes and system activities (that is, shutting down interfaces, learning of new routes, and so on), the **debug tag-switching atm-tdp states** command outputs a substantial amount of information that may interfere with system timing. Most commonly, this affects the normal operation of the Tag Distribution Protocol (TDP). You should increase the TDP hold-time value by using the **tag-switching tdp holdtime** command.

### Examples

The following is sample output from the **debug tag-switching atm-tdp states** command:

```
Router# debug tag-switching atm-tdp states

Transit Output 166.35.0.0 VPI/VCI 1/67 Active -> XmitRelease NoPath
Transit Input 166.35.0.0 VPI/VCI 1/466 Active -> ApiWaitParentLoss ParentLoss
Transit Input 166.35.0.0 VPI/VCI 1/466 ApiWaitParentLoss -> ParentWait ApiSuccess
Transit Input 166.35.0.0 VPI/VCI 1/466 ParentWait -> XmitWithdraw NoPath
Transit Input 166.35.0.0 VPI/VCI 1/466 XmitWithdraw -> XmitWithdraw Transmit
Transit Input 166.35.0.0 VPI/VCI 1/466 XmitWithdraw -> NonExistent Release
Transit Input 166.35.0.0 VPI/VCI 1/466 NonExistent -> NonExistent ApiSuccess
```

[Table 188](#) describes the significant fields shown in the display.

**Table 188** *debug tag-switching atm-tdp states Field Descriptions*

Field	Description
Transit Output	Output side of a TVC.
VPI/VCI	VC value.
Transit Input	Input side of a TVC.

# debug tag-switching packets

The **debug tag-switching packets** command is replaced by the **debug mpls packets** command. See the [debug mpls packets](#) command for more information.

# debug tag-switching tdp advertisements

To print information about the advertisement of tags and interface addresses to TDP peer devices, use the **debug tag-switching tdp advertisements** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching tdp advertisements**

**no debug tag-switching tdp advertisements**

## Syntax Description

This command has no arguments or keywords.

## Examples

The following is sample output from the **debug tag-switching tdp advertisements** command:

```
Router# debug tag-switching tdp advertisements

tagcon: adj 210.9.0.9:0 (pp 0x60D8E98C): advertise 99.101.0.8
tagcon: adj 210.9.0.9:0 (pp 0x60D8E98C): advertise 172.27.32.28
tagcon: adj 210.9.0.9:0 (pp 0x60D8E98C): advertise 10.105.0.8
tagcon: adj 210.9.0.9:0 (pp 0x60D8E98C): advertise 10.92.0.8
tagcon: adj 210.9.0.9:0 (pp 0x60D8E98C): advertise 10.205.0.8
tagcon: adj 210.9.0.9:0 (pp 0x60D8E98C): advertise 210.8.0.8
tagcon: adj 210.9.0.9:0 (pp 0x60D8E98C): advertise 10.105.0.0/16, tag 1 (#2)
tagcon: adj 210.9.0.9:0 (pp 0x60D8E98C): advertise 10.102.0.0/16, tag 26 (#4)
tagcon: adj 210.9.0.9:0 (pp 0x60D8E98C): advertise 10.227.0.0/16, tag 27 (#6)
```

[Table 189](#) describes the significant fields shown in the display.

**Table 189** *debug tag-switching tdp advertisements Field Descriptions*

Field	Description
tagcon:	Identifies the source of the message as the tag control subsystem.
adj <a.b.c.d:e>	TDP identifier of the peer device to which the advertisement has been made.
(pp 0xn timer)	Identifier for the data structure used to represent the peer device at the tag distribution level. Useful for correlating debug output.
advertise X	What was advertised to the peer device—either an interface address (“a.b.c.d”) or tag binding (“a.b.c.d/m, tag t (#n”).
(#n)	For a tag binding advertisement, the sequence number of the tag information base (TIB) modification that made it necessary to advertise the tag.

## Related Commands

Command	Description
<b>show tag-switching tdp neighbors</b>	Displays the status of TDP sessions.

## debug tag-switching tdp bindings

To print information about changes to the tag information base (TIB) used to keep track of tag bindings learned from TDP peer devices through TDP downstream tag distribution, use the **debug tag-switching tdp bindings** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching tdp bindings**

**no debug tag-switching tdp bindings**

### Syntax Description

This command has no arguments or keywords.

### Examples

The following is sample output from the **debug tag-switching tdp bindings** command:

```
Router# debug tag-switching tdp bindings

tagcon: tibent(10.105.0.0/16): created; find route tags request
tagcon: tibent(10.105.0.0/16): lcl tag 1 (#2) assigned
tagcon: tibent(10.102.0.0/16): created; find route tags request
tagcon: tibent(10.102.0.0/16): lcl tag 26 (#4) assigned
tagcon: 210.9.0.9:0: 99.101.0.9 added to addr<->tdp ident map
tagcon: 210.9.0.9:0: 172.27.32.29 added to addr<->tdp ident map
tagcon: 210.9.0.9:0: 10.105.0.9 added to addr<->tdp ident map
tagcon: tibent(172.27.32.0/22): rem tag 1 from 210.9.0.9:0 added
tagcon: tibent(200.26.0.0/16): rem tag 30 from 210.9.0.9:0 added
tagcon: tibent(210.8.0.8/32): created; remote tag learned
tagcon: tibent(210.8.0.8/32): rem tag 31 from 210.9.0.9:0 added
```

[Table 190](#) describes the significant fields shown in the display.

**Table 190** *debug tag-switching tdp bindings* Field Descriptions

Field	Description
tagcon:	Identifies the source of the message as the tag control subsystem.
tibent( <i>network/mask</i> )	Destination that has a tag binding change.
created; <i>reason</i>	TIB entry has been created for the specified destination for the indicated reason.
rem tag ...	Describes a change to the tag bindings for the specified destination. The change is for a tag binding learned from the specified TDP peer device.
lcl tag ...	Describes a change to a locally assigned (incoming) tag for the specified destination.
(# <i>n</i> )	Sequence number of the modification to the TIB corresponding to the local tag change.
<i>a.b.c.d:n: e.f.g.h</i> added to addr<->tdp ident map	Address <i>e.f.g.h</i> has been added to the set of addresses associated with TDP identifier <i>a.b.c.d:n</i> .

---

**Related Commands**

Command	Description
<b>show tag-switching tdp bindings</b>	Displays the contents of the TIB.

---

# debug tag-switching tdp directed-neighbors

To print information about the directed neighbor mechanism, use the **debug tag-switching tdp directed-neighbors** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching tdp directed-neighbors**

**no debug tag-switching tdp directed-neighbors**

## Syntax Description

This command has no arguments or keywords.

## Usage Guidelines

This mechanism establishes TDP adjacencies to peer devices that are not directly adjacent, such as peer devices at either end of a tunnel.

The directed neighbor mechanism starts TDP discovery between two TSRs that are not necessarily directly adjacent. This mechanism is used, for instance, to support two-level tagging across a TSP tunnel, and to support traffic engineering metric exchange across a TSP tunnel.

The mechanism is based on an IP address, such as the IP address of the last hop of a TSP tunnel. A TSR wanting to establish a TDP adjacency to some other TSR with a given IP address is the active TSR for that directed neighbor discovery. A TSR willing to respond to that discovery is the passive TSR for that discovery.

As with TDP discovery between adjacent TSRs, it is possible to have multiple directed neighbor discovery sessions can run between two TSRs, all supporting a single TDP adjacency.

The debug messages track discovery changes, such as discovery or loss of a directed neighbor. As a detail reflected in the debug prints, discovery of a directed neighbor with IP address X is complete when a TDP adjacency comes up and the far end announces that IP address X is one of its IP addresses.

## Examples

The following is sample output from the **debug tag-switching tdp directed-neighbors** command:

```
Router# debug tag-switching tdp directed-neighbors

tdp_directednbr: TDPDirAdj 10.11.10.11 received address addition notification
tdp_directednbr: TDPDirAdj 10.11.10.11 TDP peer set
tdp_directednbr: TDPDirAdj 10.11.10.11 received address deletion notification
tdp_directednbr: TDPDirAdj 10.11.10.11 peer cleared
```

[Table 191](#) describes the significant fields shown in the display.

**Table 191** *debug tag-switching tdp directed-neighbors Field Descriptions*

Field	Description
tdp_directednbr:	Identifies this as a TDP directed neighbor debug statement.
TDPDirAdj <address>	Identifies the IP address to which a TDP adjacency is desired.

## Related Commandss

Command	Description
<b>show tag-switching tdp neighbors</b>	Displays the status of TDP sessions.

# debug tag-switching tdp peer state-machine

To print information about state transitions at the tag distribution level, use the **debug tag-switching tdp peer state-machine** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching tdp peer state-machine**

**no debug tag-switching tdp peer state-machine**

## Syntax Description

This command has no arguments or keywords.

## Usage Guidelines

TDP sessions are supported by data structures and state machines at three levels:

- Transport—The transport level establishes and maintains TCP connections used to support TDP sessions.
- Protocol—The protocol level implements the TDP session setup protocol, and constructs and parses TDP PDUs and PIEs.
- Tag distribution—The tag distribution level uses TDP sessions to exchange tags with TDP peer devices.

The **debug tag-switching tdp transport** command provides visibility of activity at the transport level, the **debug tag-switching tdp session** command at the protocol level, and the **debug tag-switching tdp peer** command at the tag distribution level.

## Examples

The following is sample output from the **debug tag-switching tdp peer state-machine** command:

```
Router# debug tag-switching tdp peer state-machine

tagcon: start TDP TCP timers for 202.0.0.1:1 (pp 0x60D8ABC8)
tagcon: adj 202.0.0.1:1-1 (pp 0x60D8ABC8): Event unsol open
      unsol op pdg -> estab
tagcon: start TDP TCP timers for 210.9.0.9:0 (pp 0x60D93608)
tagcon: adj 210.9.0.9:0 (pp 0x60D93608): Event unsol open
      unsol op pdg -> estab
tagcon: adj 210.9.0.9:0 (pp 0x60D93608): Event down
      estab -> dstroy
tagcon: adj 202.0.0.1:1 (pp 0x60D8ABC8): Event down
      estab -> dstroy
tagcon: start TDP TCP timers for 202.0.0.1:1 (pp 0x60DAC678)
tagcon: adj 202.0.0.1:1-1 (pp 0x60DAC678): Event unsol open
      unsol op pdg -> defrd
tagcon: start TDP TCP timers for 210.9.0.9:0 (pp 0x60D895C4)
tagcon: adj 210.9.0.9:0 (pp 0x60D895C4): Event unsol open
      unsol op pdg -> defrd
tagcon: adj 210.9.0.9:0 (pp 0x60D93608): Event cleanup done
      dstroy -> non-ex
tagcon: adj 210.9.0.9:0 (pp 0x60D895C4): Event undefer
      defrd -> estab
tagcon: adj 202.0.0.1:1 (pp 0x60D8ABC8): Event cleanup done
      dstroy -> non-ex
tagcon: adj 202.0.0.1:1-1 (pp 0x60DAC678): Event undefer
      defrd -> estab
```

Table 192 describes the significant fields shown in the display.

**Table 192** *debug tag-switching tdp peer state-machine Field Descriptions*

Field	Description
tagcon:	Identifies the source of the message as the tag control subsystem.
adj <i>a.b.c.d:e</i>	TDP identifier of the peer device for the session with the state change.
(pp 0xnnnnnnnn)	Address of the data structure used to represent the peer device at the tag distribution level. It is useful for correlating debug output.
Event E	Event causing the state change.
S1 -> S2	State of the TDP session has changed from state S1 to state S2.

## debug tag-switching tdp pies received

To print information about TDP protocol information elements (PIEs) received from TDP peer devices, use the **debug tag-switching tdp pies received** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching tdp pies received [all]**

**no debug tag-switching tdp pies received [all]**

### Syntax Description

**all** (Optional) TDP received PIEs, including periodic keepalive PIEs.

### Usage Guidelines

TDP requires periodic transmission of keepalive PIEs. If you do not specify the **all** option, periodic keepalive PIEs are not displayed.

### Examples

The following is sample output from the **debug tag-switching tdp pies received** command:

```
Router# debug tag-switching tdp pies received all

tdp: Rcvd open PIE from 202.0.0.1 (pp 0x0)
tdp: Rcvd keep_alive PIE from 202.0.0.1:1 (pp 0x0)
tdp: Rcvd request_bind PIE from 202.0.0.1:1 (pp 0x60DAC678)
tdp: Rcvd request_bind PIE from 202.0.0.1:1 (pp 0x60DAC678)
tdp: Rcvd open PIE from 210.9.0.9 (pp 0x0)
tdp: Rcvd keep_alive PIE from 210.9.0.9:0 (pp 0x0)
tdp: Rcvd bind PIE from 202.0.0.1:1 (pp 0x60DAC678)
tdp: Rcvd bind PIE from 202.0.0.1:1 (pp 0x60DAC678)
```

[Table 193](#) describes the significant fields shown in the display.

**Table 193** *debug tag-switching tdp pies received all Field Descriptions*

Field	Description
tdp:	Identifies the source of the message as TDP.
Rcvd <i>xxx</i> PIE	Type of PIE received.
from <i>a.b.c.d</i>	Host that sent the PIE. Used in the early stages of the opening of a TDP session, when the TDP identifier is not yet known.
from a.b.c.d:e	TDP identifier of the peer device that sent the PIE.
(pp 0xnnnnnnnn)	Identifies the data structure used to represent the peer device at the tag distribution level. Useful for correlating debug output.

### Related Commands

Command	Description
<a href="#">debug tag-switching tdp pies sent</a>	Prints information about state transitions at the tag distribution level.

# debug tag-switching tdp pies sent

To print information about state transitions at the tag distribution level, use the **debug tag-switching tdp pies sent** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching tdp pies sent [all]**

**no debug tag-switching tdp pies sent [all]**

## Syntax Description

**all** (Optional) TDP sent PIEs, including periodic keepalive PIEs.

## Usage Guidelines

TDP requires periodic transmission of keepalive PIEs. If you do not specify the **all** option, periodic keepalive PIEs are not displayed.

## Examples

The following is sample output from the **debug tag-switching tdp pies sent all** command:

```
Router# debug tag-switching tdp pies sent all

tdp: Queued open PIE to 210.222.0.222:1 (pp 0x0)
tdp: Sent open PIE to 210.222.0.222:1 (pp 0x0)
tdp: Queued keep_alive PIE to 210.222.0.222:1 (pp 0x0)
tdp: Sent keep_alive PIE to 210.222.0.222:1 (pp 0x0)
tdp: Queued request_bind PIE to 210.222.0.222:1 (pp 0x60F264C8)
tdp: Sent request_bind PIE to 210.222.0.222:1 (pp 0x60F264C8)
tdp: Queued request_bind PIE to 210.222.0.222:1 (pp 0x60F264C8)
tdp: Sent request_bind PIE to 210.222.0.222:1 (pp 0x60F264C8)
tdp: Queued open PIE to 210.8.0.8 (pp 0x0)
tdp: Queued bind PIE to 210.222.0.222:1 (pp 0x60F264C8)
tdp: Sent bind PIE to 210.222.0.222:1 (pp 0x60F264C8)
tdp: Queued bind PIE to 210.222.0.222:1 (pp 0x60F264C8)
tdp: Sent bind PIE to 210.222.0.222:1 (pp 0x60F264C8)
tdp: Queued bind PIE to 210.222.0.222:1 (pp 0x60F264C8)
tdp: Queued open PIE to 210.8.0.8 (pp 0x0)
tdp: Sent open PIE to 210.8.0.8 (pp 0x0)
tdp: Queued keep_alive PIE to 210.8.0.8:0 (pp 0x0)
tdp: Sent keep_alive PIE to 210.8.0.8:0 (pp 0x0)
tdp: Queued address PIE to 210.8.0.8:0 (pp 0x60F161AC)
tdp: Sent address PIE to 210.8.0.8:0 (pp 0x60F161AC)
tdp: Queued bind PIE to 210.8.0.8:0 (pp 0x60F161AC)
tdp: Queued bind PIE to 210.8.0.8:0 (pp 0x60F161AC)
tdp: Queued bind PIE to 210.8.0.8:0 (pp 0x60F161AC)
tdp: Queued bind PIE to 210.8.0.8:0 (pp 0x60F161AC)
tdp: Queued bind PIE to 210.8.0.8:0 (pp 0x60F161AC)
tdp: Sent bind PIE to 210.8.0.8:0 (pp 0x60F161AC)
tdp: Sent bind PIE to 210.8.0.8:0 (pp 0x60F161AC)
tdp: Sent bind PIE to 210.8.0.8:0 (pp 0x60F161AC)
tdp: Sent bind PIE to 210.8.0.8:0 (pp 0x60F161AC)
tdp: Sent bind PIE to 210.8.0.8:0 (pp 0x60F161AC)
```

Table 194 describes the significant fields shown in the display.

**Table 194** *debug tag-switching tdp sent all Field Descriptions*

Field	Description
tdp:	Identifies the source of the message as TDP.
Queued <i>xxx</i> PIE	Indicates that a PIE of the specified type has been queued for transmission.
Sent <i>xxx</i> PIE	Indicates that a PIE of the specified type has been sent on the TDP session TCP connection.
to <i>a.b.c.d</i>	Host to which the PIE has been sent or for which it has been queued. Used in the early stages of opening a TDP session when the TDP identifier is not yet known.
to <i>a.b.c.d:e</i>	TDP identifier of the peer device to which the PIE has been sent or for which it has been queued.
(pp 0xnnnnnnnn)	Identifies the data structure used to represent the peer device at the tag distribution level. Useful for correlating debug output.

#### Related Commands

Command	Description
<b>debug tag-switching tdp pies received</b>	Prints information about TDP PIEs received from TDP peer devices.
<b>debug tag-switching tdp session io</b>	Prints the contents of TDP PIEs sent to and received from TDP peer devices.

# debug tag-switching tdp session io

To print the contents of TDP PIEs sent to and received from TDP peer devices, use the **debug tag-switching tdp session io** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching tdp session io [all]**

**no debug tag-switching tdp session io [all]**

## Syntax Description

**all** (Optional) TDP session I/O activity, including I/O for periodic keepalives.

## Usage Guidelines

TDP sessions are supported by data structures and state machines at three levels:

- Transport—The transport level establishes and maintains TCP connections used to support TDP sessions.
- Protocol—The protocol level implements the TDP session setup protocol, and constructs and parses TDP PDUs and PIEs.
- Tag distribution—The tag distribution level uses TDP sessions to exchange tags with TDP peer devices.

The **debug tag-switching tdp transport** command provides visibility of activity at the transport level, the **debug tag-switching tdp session** command at the protocol level, and the **debug tag-switching tdp peer** command at the tag distribution level.

TDP requires periodic transmission of keepalive PIEs. If you do not specify the **all** option, periodic keepalive PIEs are not displayed.

## Examples

The following is sample output from the **debug tag-switching tdp session io all** command:

```
Router# debug tag-switching tdp session io all

tdp: Rcvd open PIE from 210.9.0.9 (pp 0x0)
tdp: TDP open PIE: PDU hdr: TDP Id: 210.9.0.9:0; PIE Contents:
 0x00 0x01 0x00 0x10 0xD2 0x09 0x00 0x09 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x04
 0x01 0x00 0x00 0x1E
tdp: Sent open PIE to 210.9.0.9:0 (pp 0x0)
tdp: TDP open PIE: PDU hdr: TDP Id: 172.27.32.28:0; PIE Contents:
 0x00 0x01 0x00 0x10 0xAC 0x1B 0x20 0x1C 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x04
 0x01 0x00 0x00 0x0F
tdp: Sent keep_alive PIE to 210.9.0.9:0 (pp 0x0)
tdp: TDP keep_alive PIE: PDU hdr: TDP Id: 172.27.32.28:0; PIE Contents:
 0x00 0x01 0x00 0x0C 0xAC 0x1B 0x20 0x1C 0x00 0x00 0x00 0x00 0x05 0x00 0x00 0x00
tdp: Rcvd keep_alive PIE from 210.9.0.9:0 (pp 0x0)
tdp: TDP keep_alive PIE: PDU hdr: TDP Id: 210.9.0.9:0; PIE Contents:
 0x00 0x01 0x00 0x0C 0xD2 0x09 0x00 0x09 0x00 0x00 0x00 0x00 0x05 0x00 0x00 0x00
tdp: Rcvd address PIE from 210.9.0.9:0 (pp 0x60E109F0)
tdp: TDP address PIE: PDU hdr: TDP Id: 210.9.0.9:0; PIE Contents:
 0x00 0x01 0x00 0x35 0xD2 0x09 0x00 0x09 0x00 0x00 0x00 0x00 0x08 0x00 0x00 0x29
 0x00 0x01 0x00 0x03 0x00 0x23 0x20 0x63 0x65 0x00 0x09 0x20 0xAC 0x1B 0x20 0x1D
 0x20 0x0A 0x69 0x00 0x09 0x20 0x0A 0x5C 0x00 0x09 0x20 0x0A 0x6F 0x00 0x09 0x20
 0x0A 0xCD 0x00 0x09 0x20 0xD2 0x09 0x00 0x09
tdp: Rcvd bind PIE from 210.9.0.9:0 (pp 0x60E109F0)
```

```

tdp: TDP bind PIE: PDU_hdr: TDP Id: 210.9.0.9:0; PIE Contents:
0x00 0x01 0x00 0xFC 0xD2 0x09 0x00 0x09 0x00 0x00 0x00 0x02 0x00 0x00 0xF0
0x00 0x00 0x00 0x00 0x00 0x01 0x00 0x02 0x00 0xE6 0x00 0x00 0x00 0x01 0x10
0x0A 0x6F 0x00 0x00 0x00 0x00 0x01 0x16 0xAC 0x1B 0x20 0x00 0x00 0x00 0x01
0x10 0xD2 0x09 0x00 0x00 0x00 0x00 0x1A 0x20 0x0A 0x0B 0x00 0x0B 0x00 0x00

```

Table 195 describes the significant fields shown in the display.

**Table 195** *debug tag-switching tdp session io Field Descriptions*

Field	Description
tdp:	Identifies the source of the message as TDP.
Rcvd <i>xxx</i> PIE	Indicates that a PIE of the specified type has been received.
from <i>a.b.c.d</i>	Host to which the PIE has been sent. Used in the early stages of the opening of a TDP session when the TDP identifier is not yet known.
Sent <i>xxx</i> PIE	Indicates that a PIE of the specified type has been sent.
to <i>a.b.c.d</i>	Host to which the PIE has been sent. Used in the early stages of opening a TDP session when the TDP identifier is not yet known.
to <i>a.b.c.d:e</i>	TDP identifier of the peer device to which the PIE has been sent.
(pp 0xnnnnnnnn)	Identifies the data structure used to represent the peer device at the tag distribution level. Useful for correlating debug output.
--TDP <i>xxx</i> PIE	Type of PIE that has been sent.
PDU_hdr: TDP Id: <i>a.b.c.d:e</i>	TDP identifier of the sender included in the TDP PDU header.
PIE contents: 0xnn ... 0xnn	Contents of the PIE represented as a sequence of bytes.

#### Related Commands

Command	Description
<a href="#">debug tag-switching tdp pies received</a>	Prints information about TDP PIEs received from TDP peer devices.
<a href="#">debug tag-switching tdp pies sent</a>	Prints information about state transitions at the tag distribution level.

# debug tag-switching tdp session state-machine

To print information about state transitions at the protocol level, use the **debug tag-switching tdp session state-machine** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching tdp session state-machine**

**no debug tag-switching tdp session state-machine**

**Syntax Description** This command has no arguments or keywords.

**Usage Guidelines** TDP sessions are supported by data structures and state machines at three levels:

- Transport—The transport level establishes and maintains TCP connections used to support TDP sessions.
- Protocol—The protocol level implements the TDP session setup protocol, and constructs and parses TDP PDUs and PIEs.
- Tag distribution—The tag distribution level uses TDP sessions to exchange tags with TDP peer devices.

The **debug tag-switching tdp transport** command provides visibility of activity at the transport level, the **debug tag-switching tdp session** command at the protocol level, and the **debug tag-switching tdp peer** command at the tag distribution level.

**Examples** The following is sample output from the **debug tag-switching tdp session state-machine** command:

```
Router# debug tag-switching tdp session state-machine

tdp: adj:210.9.0.9(0x60DDBB4C): Event: Xport opened;
      Non-existent -> Init pasv
tdp: tdp_create_ptcl_adj: tp = 0x60DDBB4C, ipaddr = 210.9.0.9
tdp: adj:210.9.0.9(0x60DDBB4C): Event: Xport opened;
      Init pasv -> Init pasv
tdp: adj:10.105.0.9(0x60DDBB4C): Event: Rcv TDP Open;
      Init pasv -> Open rcvd pasv
tdp: adj:10.105.0.9(0x60DDBB4C): Event: Rcv TDP KA;
      Open rcvd pasv -> Oper
tdp: adj:unknown(0x60DDBB4C): Event: Xport closed;
      Oper -> Non-existent
```

Table 196 describes the significant fields shown in the display.

**Table 196** *debug tag-switching tdp session state-machine Field Descriptions*

Field	Description
tdp:	Identifies the source of the message as TDP.
adj: <i>a.b.c.d</i>	Identifies the network address of the TDP peer device.
( <i>0xnnnnnnnn</i> )	Identifies the data structure used to represent the peer device at the protocol level. Useful for correlating debug output.

**Table 196** *debug tag-switching tdp session state-machine Field Descriptions (continued)*

Field	Description
Event: E	Event that caused the state transition.
S1 -> S2	State of the TDP session has changed from state S1 to state S2.

# debug tag-switching tdp transport connections

To print information about the TCP connections used to support TDP sessions, use the **debug tag-switching tdp transport connections** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching tdp transport connections**

**no debug tag-switching tdp transport connections**

**Syntax Description** This command has no arguments or keywords.

**Usage Guidelines** TDP sessions are supported by data structures and state machines at three levels:

- Transport—The transport level establishes and maintains TCP connections used to support TDP sessions.
- Protocol—The protocol level implements the TDP session setup protocol, and constructs and parses TDP PDUs and PIEs.
- Tag distribution—The tag distribution level uses TDP sessions to exchange tags with TDP peer devices.

The **debug tag-switching tdp transport** command provides visibility of activity at the transport level, the **debug tag-switching tdp session** command at the protocol level, and the **debug tag-switching tdp peer** command at the tag distribution level.

When two devices establish a TCP connection for a TDP session, the device with the larger transport address plays an active role and the other plays a passive role. The active device attempts to establish a TCP connection to the well-known TDP port at the passive device. The passive device waits for the connection to the well-known port to be established.

**Examples** The following is sample output from the **debug tag-switching transport connections** command:

```
Router# debug tag-switching tdp transport connections

Debug output at active peer:

tdp: Opening conn; adj 0x60F7C604, 210.9.0.9 <-> 172.27.32.28
tdp: Conn is up; adj 0x60F7C604, 210.9.0.9:11018 <-> 172.27.32.28:711
tdp: hold-timer expired for adj 0x60F7C604, will close conn
tdp: Closing conn 210.9.0.9:11018 <-> 172.27.32.28:711, adj 0x60F7C604

Debug output at passive peer:

tdp: Incoming conn 172.27.32.28:711 <-> 210.9.0.9:11018
tdp: Conn closed by peer; adj 0x60EB5FD4
      172.27.32.28:711 <-> 210.9.0.9:11018, Ethernet1/1/1
tdp: Closing conn 172.27.32.28:711 <-> 210.9.0.9:11018, adj 0x60EB5FD4
```

Table 197 describes the significant fields shown in the display.

**Table 197** *debug tag-switching tdp transport connections Field Descriptions*

Field	Description
tdp:	Identifies the source of the message as TDP.
adj 0xnnnnnnnn	Identifies the data structure used to represent the peer device at the transport level. Useful for correlating debug output.
<i>a.b.c.d -&gt; p.q.r.s</i>	Indicates a TCP connection between a.b.c.d and p.q.r.s.
<i>a.b.c.d:x -&gt; p.q.r.s:y</i>	Indicates a TCP connection between a.b.c.d, port x and p.q.r.s, port y.

#### Related Commands

Command	Description
<b>debug tag-switching tdp transport events</b>	Prints information about the events related to the TDP peer discovery mechanism, which is used to determine the devices with which to establish TDP sessions.

# debug tag-switching tdp transport events

To print information about the events related to the TDP peer discovery mechanism, which is used to determine the devices with which to establish TDP sessions, use the **debug tag-switching tdp transport events** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching tdp transport events**

**no debug tag-switching tdp transport events**

## Syntax Description

This command has no arguments or keywords.

## Usage Guidelines

TDP sessions are supported by data structures and state machines at three levels:

- Transport—The transport level establishes and maintains TCP connections used to support TDP sessions.
- Protocol—The protocol level implements the TDP session setup protocol, and constructs and parses TDP PDUs and PIEs.
- Tag distribution—The tag distribution level uses TDP sessions to exchange tags with TDP peer devices.

The **debug tag-switching tdp transport** command provides visibility of activity at the transport level, the **debug tag-switching tdp session** command at the protocol level, and the **debug tag-switching tdp peer** command at the tag distribution level.

## Examples

The following is sample output from the **debug tag-switching tdp transport events** command:

```
Router# debug tag-switching tdp transport events

tdp: Rcvd hello; Ethernet1/1/1, from 10.105.0.9 (210.9.0.9:0), intf_id 0, opt 0x4
tdp: Hello from 10.105.0.9 (210.9.0.9:0) to 255.255.255.255, opt 0x4
tdp: New adj 0x60DF6E50 from 10.105.0.9 (210.9.0.9:0), Ethernet1/1/1
tdp: Rcvd hello; ATM3/0.1, from 200.26.0.4 (202.0.0.1:1), intf_id 1, opt 0x4, tcatm
tdp: Rcvd hello; Ethernet1/1/1, from 10.105.0.9 (210.9.0.9:0), intf_id 0, opt 0x4
tdp: Hello from 10.105.0.9 (210.9.0.9:0) to 255.255.255.255, opt 0x4
tdp: Ignore Hello Timer for Ethernet1/1/1; intf not TDP ready
tdp: Send hello; Ethernet1/1/1, src/dst 10.105.0.8/255.255.255.255, inst_id 0
tdp: Incoming conn 172.27.32.28:711 <-> 210.9.0.9:11019
tdp: Found adj 0x60DF6E50 for 210.9.0.9 (Hello xport addr opt)
tdp: New temporary adj 0x61033D38 from 210.9.0.9
tdp: Real adj 0x60DF6E50 bound to 210.9.0.9:0, replacing temp adj 0x61033D38
tdp: Adj 0x61033D38; state set to closed
tdp: Rcvd hello; Ethernet1/1/1, from 10.105.0.9 (210.9.0.9:0), intf_id 0, opt 0x4
tdp: Rcvd hello; ATM3/0.1, from 200.26.0.4 (202.0.0.1:1), intf_id 1, opt 0x4, tcatm
tdp: Send hello; ATM3/0.1, src/dst 99.101.0.8/255.255.255.255, inst_id 1, tcatm
tdp: Rcvd hello; Ethernet1/1/1, from 10.105.0.9 (210.9.0.9:0), intf_id 0, opt 0x4
tdp: Send hello; Ethernet1/1/1, src/dst 10.105.0.8/255.255.255.255, inst_id 0
tdp: Rcvd hello; ATM3/0.1, from 200.26.0.4 (202.0.0.1:1), intf_id 1, opt 0x4, tcatm
```

Table 198 describes the significant fields shown in the display.

**Table 198** *debug tag-switching tdp transport events Field Descriptions*

Field	Description
tdp:	Identifies the source of the message as TDP.
adj 0xnnnnnnnn	Identifies the data structure used to represent the peer device at the transport level. Useful for correlating debug output.
a.b.c.d (p.q.r.s:n)	Network address and TDP identifier of the peer device.
intf_id	Interface identifier (nonzero for TC-ATM interfaces, 0 otherwise).
opt 0xn	Bits that describe options in the TDP discovery hello packet: <ul style="list-style-type: none"> <li>• 0x1—Directed hello option</li> <li>• 0x2—Send directed hello option</li> <li>• 0x4—Transport address option</li> </ul>

#### Related Commands

Command	Description
<a href="#">debug tag-switching tdp transport connections</a>	Prints information about the TCP connections used to support TDP sessions.

# debug tag-switching tdp transport timers

To print information about events that restart the “hold” timers that are part of the TDP discovery mechanism, use the **debug tag-switching tdp transport timers** privileged EXEC command. The **no** form of this command disables debugging output.

**debug tag-switching tdp transport timers**

**no debug tag-switching tdp transport timers**

## Syntax Description

This command has no arguments or keywords.

## Usage Guidelines

TDP sessions are supported by data structures and state machines at three levels:

- Transport—The transport level establishes and maintains TCP connections used to support TDP sessions.
- Protocol—The protocol level implements the TDP session setup protocol. The construction and parsing of TDP PDUs and PIEs occur at this level.
- Tag distribution—The tag distribution level uses TDP sessions to exchange tags with TDP peer devices.

The **debug tag-switching tdp transport** command provides visibility of activity at the transport level, the **debug tag-switching tdp session** command at the protocol level, and the **debug tag-switching tdp peer** command at the tag distribution level.

## Examples

The following is sample output from the **debug tag-switching tdp transport timers** command:

```
Router# debug tag-switching tdp transport timers

tdp: Start holding timer; adj 0x60D5BC10, 200.26.0.4
tdp: Start holding timer; adj 0x60EA9360, 10.105.0.9
tdp: Start holding timer; adj 0x60D5BC10, 200.26.0.4
tdp: Start holding timer; adj 0x60EA9360, 10.105.0.9
tdp: Start holding timer; adj 0x60D5BC10, 200.26.0.4
tdp: Start holding timer; adj 0x60EA9360, 10.105.0.9
```

[Table 199](#) describes the significant fields shown in the display.

**Table 199** *debug tag-switching tdp transport timers Field Descriptions*

Field	Description
tdp	Identifies the source of the message as TDP.
adj 0xn timer	Identifies the data structure used to represent the peer device at the transport level.
a.b.c.d	Network address of the peer device.

---

**Related Commands**

Command	Description
<a href="#">debug tag-switching tdp transport events</a>	Prints information about the events related to the TDP peer discovery mechanism, which is used to determine the devices with which to establish TDP sessions.

---

## debug tag-switching tfib cef

The **debug tag-switching tfib cef** command is replaced by the **debug mpls lfib cef** command. See the [debug mpls lfib cef](#) command for more information.

## debug tag-switching tfib enc

The **debug tag-switching tfib enc** command is replaced by the **debug mpls lfib enc** command. See the [debug mpls lfib enc](#) command for more information.

## debug tag-switching tfib state

The **debug tag-switching tfib state** command is replaced by the **debug mpls lfib state** command. See the [debug mpls lfib state](#) command for more information.

## debug tag-switching tfib struct

The **debug tag-switching tfib struct** command is replaced by the **debug mpls lfib struct** command. See the [debug mpls lfib struct](#) command for more information.

## debug tag-switching tfib tsp

The **debug tag-switching tfib tsp** command is replaced by the **debug mpls lfib lsp** command. See the [debug mpls lfib lsp](#) command for more information.

## debug tag-switching tsp-tunnels events

The **debug tag-switching tsp-tunnels events** command is replaced by the **debug mpls traffic-eng tunnels events** command. See the [debug mpls traffic-eng tunnels events](#) command for more information.

## debug tag-switching tsp-tunnels signalling

The **debug tag-switching tsp-tunnels signalling** command is replaced by the **debug mpls traffic-eng tunnels signalling** command. See the [debug mpls traffic-eng tunnels signalling](#) command for more information.

## debug tag-switching tsp-tunnels tagging

The **debug tag-switching tsp-tunnels tagging** command is replaced by the **debug mpls traffic-eng tunnels labels** command. See the [debug mpls traffic-eng tunnels labels](#) command for more information.

# debug tag-switching xtagatm cross-connect

To display requests and responses for establishing and removing cross-connects on the controlled ATM switch, use the **debug tag-switching xtagatm cross-connect** command. The **no** form of this command disables debugging output.

**debug tag-switching xtagatm cross-connect**

**no debug tag-switching xtagatm cross-connect**

**Syntax Description** This command has no arguments or keywords.

**Defaults** No default behavior or values.

Command History	Release	Modification
	12.0(5)T	This command was introduced.

**Usage Guidelines** Use the **debug tag-switching xtagatm cross-connect** command to monitor requests to establish or remove cross-connects from XTagATM interfaces to the VSI master, and the VSI master's responses to these requests.



**Note**

Use this command with care, because it generates output for each cross-connect operation performed by the LSC. In a network configuration with a large number of label virtual circuits (LVCs), the volume of output generated may interfere with system timing and the proper operation of other router functions. Use this command only in situations in which the LVC setup or teardown rate is low.

**Examples** The following is sample output from the **debug tag-switching xtagatm cross-connect** command:

```
Router# debug tag-switching xtagatm cross-connect

XTagATM: cross-conn request; SETUP, userdata 0x17, userbits 0x1, prec 7
        0xC0100 (Ct1-If) 1/32 <-> 0xC0200 (XTagATM0) 0/32
XTagATM: cross-conn response; DOWN, userdata 0x60CDCB5C, userbits 0x2, result
OK
        0xC0200 1/37 --> 0xC0300 1/37
```

Table 200 describes the significant fields shown in the sample command output shown above.

**Table 200** *debug tag-switching xtagatm cross-connect Field Descriptions*

Field	Description
XTagATM	Identifies the source of the debug message as an XTagATM interface.
cross-conn	Indicates that the debug message pertains to a cross-connect setup or teardown operation.
request	A request from an XTagATM interface to the VSI master to set up or tear down a cross-connect.
response	Response from the VSI master to an XTagATM interface that a cross-connect was set up or removed.
SETUP	A request for the setup of a cross-connect.
TEARDOWN	A request for the teardown of a cross-connect.
UP	The cross-connect is established.
DOWN	The cross-connect is not established.
userdata, userbits	Values passed with the request that are returned in the corresponding fields shown in the matching response.
prec	The precedence for the cross-connect.
result	Indicates the status of the completed request.
0xC0100 (Ctl-If) 1/32	Indicates the following: that one endpoint of the cross-connect is on the interface whose logical interface number is 0xC0100; that this interface is the VSI control interface; that the VPI value at this endpoint is 1; and that the VCI value at this end of the cross-connect is 32.
<->	Indicates that this is a bidirectional cross-connect.
0xC0200 (XTagATM0) 0/32	Indicates the following: that the other endpoint of the cross-connect is on the interface whose logical interface number is 0xC0200; that this interface is associated with XTagATM interface 0; that the VPI value at this endpoint is 0; and that the VCI value at this end of the cross-connect is 32.
->	Indicates that this response pertains to a unidirectional cross-connect.

#### Related Commands

Command	Description
<b>show xtagatm cross-connect</b>	Displays information about remotely connected ATM switches.

# debug tag-switching xtagatm errors

To display information about error and abnormal conditions that occur on XTagATM interfaces, use the **debug tag-switching xtagatm errors** command. The **no** form of this command disables debugging output.

**debug tag-switching xtagatm errors**

**no debug tag-switching xtagatm errors**

---

**Syntax Description** This command has no arguments or keywords.

---

**Defaults** No default behavior or values.

---

Command History	Release	Modification
	12.0(5)T	This command was introduced.

---



---

**Usage Guidelines** Use the **debug tag-switching xtagatm errors** command to display information about abnormal conditions and events that occur on XTagATM interfaces.

---

**Examples** The following is sample output from the **debug tag-switching xtagatm errors** command:

```
Router# debug tag-switching xtagatm errors
```

```
XTagATM VC: XTagATM0 1707 2/352 (ATM1/0 1769 3/915): Cross-connect setup
failed NO_RESOURCES
```

This message indicates that an attempt to set up a cross-connect for a terminating VC on XTagATM interface 0 failed, and that the reason for the failure was a lack of resources on the controlled ATM switch.

# debug tag-switching xtagatm events

To display information about major events that occur on XTagATM interfaces, not including events for specific XTagATM VCs and switch cross-connects, use the following **debug tag-switching xtagatm events** command. The **no** form of this command disables debugging output.

**debug tag-switching xtagatm events**

**no debug tag-switching xtagatm events**

**Syntax Description** This command has no arguments or keywords.

**Defaults** No default behavior or values.

Command History	Command	Modification
	12.0(5)T	This command was introduced.

**Usage Guidelines** Use the **debug tag-switching xtagatm events** command to monitor major events that occur on XTagATM interfaces. This command monitors only events that pertain to XTagATM interfaces as a whole and does not include any events that pertain to individual XTagATM VCs or individual switch cross-connects. The specific events monitored when the **debug tag-switching xtagatm events** command is in effect include the following:

- Receipt of asynchronous notifications sent by the VSI master through the external ATM API (ExATM API) to an XTagATM interface.
- Resizing of the table that is used to store switch cross-connect information. This table is resized automatically as the number of cross-connects increases.
- Marking of XTagATM VCs as stale when an XTagATM interface shuts down, thereby ensuring that the stale interfaces are refreshed before new XTagATM VCs can be created on the interface.

**Examples** The following is sample output from the **debug tag-switching xtagatm events** command:

```
Router# debug tag-switching xtagatm events

XTagATM: desired cross-connect table size set to 256
XTagATM: ExATM API intf event Up, port 0xA0100 (None)
XTagATM: ExATM API intf event Down, port 0xA0100 (None)
XTagATM: marking all VCs stale on XTagATM0
```

Table 201 describes the significant fields shown in the sample command output shown above.

**Table 201** *debug tag-switching xtagatm events Field Descriptions*

Field	Description
XTagATM	Identifies the source of the debug message as an XTagATM interface.
desired cross-connect table size set to 256	Indicates that the table of cross-connect information has been set to hold 256 entries. A single cross-connect table is shared among all XTagATM interfaces. The cross-connect table is automatically resized as the number of cross-connects increases.
ExATM API	Indicates that the information in the debug output pertains to an asynchronous notification sent by the VSI master to the XTagATM driver.
event Up/Down	Indicates the specific event that was sent by the VSI master to the XTagATM driver.
port 0xA0100 (None)	Indicates that the event pertains to the VSI interface whose logical interface number is 0xA0100, and that this logical interface is not bound (through the <b>extended-port</b> interface configuration command) to any XTagATM interface.
marking all VCs stale on XTagATM0	Indicates that all existing XTagATM VCs on interface XTagATM0 are marked as stale, and that XTagATM0 remains down until all of these VCs are refreshed.

# debug tag-switching xtagatm vc

To display information about events that affect individual XTagATM terminating VCs, use the **debug tag-switching xtagatm vc** command. The **no** form of this command disables debugging output.

**debug tag-switching xtagatm vc**

**no debug tag-switching xtagatm vc**

**Syntax Description** This command has no arguments or keywords.

**Defaults** No default behavior or values.

Command History	Release	Modification
	12.0(5)T	This command was introduced.

**Usage Guidelines** Use the **debug tag-switching xtagatm vc** command to display detailed information about all events that affect individual XTagATM terminating VCs.



**Note**

Use this command with care, because it results in extensive output when many XTagATM VCs are set up or torn down. This output can interfere with system timing and normal operation of other router functions. Use the **debug tag-switching xtagatm vc** command only when a few XTagATM VCs are created or removed.

**Examples** The following is sample output from the **debug tag-switching xtagatm vc** command:

```
Router# debug tag-switching xtagatm vc

XTagATM VC: XTagATM1 18 0/32 (ATM1/0 0 0/0): Setup, Down --> UpPend
XTagATM VC: XTagATM1 18 0/32 (ATM1/0 88 1/32): Complete, UpPend --> Up
XTagATM VC: XTagATM1 19 1/33 (ATM1/0 0 0/0): Setup, Down --> UpPend
XTagATM VC: XTagATM0 43 0/32 (ATM1/0 67 1/84): Teardown, Up --> DownPend
```

Table 202 describes the significant fields shown in the display.

**Table 202** *debug tag-switching stagatm vc Field Descriptions*

Field	Description
XTagATM VC	Identifies the source of the debug message as the XTagATM interface terminating VC facility.
XTagATM <ifnum>	Identifies the particular XTagATM interface number for the terminating VC.
vcd vpi/vci	Indicates the VCD and VPI/VCI values for the terminating VC.
(ctl-if vcd vpi/vci)	Indicates the control interface, the VCD, and the VPI and VCI values for the private VC corresponding to the XTagATM VC on the control interface.
Setup, Complete, Teardown	Indicates the name of the particular event that has occurred for the indicated VC.
oldstate -> newstate	Indicates the state of the terminating VC before and after the processing of the indicated event.