

Tag Switch Controller

Feature Summary

The Tag Switch Controller (TSC) with Cisco's BPX 8620 wide area switch and BPX 8650 IP+ATM switch delivers scalable integration of IP services over an ATM network.

The TSC enables the BPX 8620 and 8650 to:

- Participate in a Tag Switching network
- Directly peer with IP edge routers
- Support the full suite of IP features available in Cisco IOS

Tag Switching's highly scalable IP+ATM integration is created by the TSC using a direct peer relationship between the BPX 8620 or 8650 and IP edge routers. This removes the limit placed on the number of IP edge routers, seen in traditional IP over ATM networks, thus allowing service providers to keep pace with the growing demand for IP services. The TSC additionally supports the easy, quick, and direct implementation of advanced IP services over an ATM network of BPX 8620s and 8650s.

Tag Switching combines the performance and virtual circuit capabilities of Layer 2 (data link layer) switching with the proven scalability of Layer 3 (network layer) routing to deliver a solution to service providers that meets the challenge of managing explosive growth and providing differentiated services while leveraging their existing infrastructure. The Tag Switching architecture is unique in providing the flexibility to run over any combination of Layer 2 technologies and support any Layer 3 protocol while scaling well beyond today's current solutions. Thus deploying Tag Switching across an Internet or large enterprise networks, customers can save money by using existing ATM and routing infrastructures, grow revenue with tag-enabled services, and increase productivity with enhanced network scalability and performance.

List of Terms

The following terms are defined for a Tag Switching context only, not for general situations.

ATM-TSR—A Tag Switching router with a number of TC-ATM interfaces. The router forwards the cells from these interfaces using tags carried in the VPI and/or VCI field.

ATM edge TSR—A Tag Switching router that is connected to the ATM-TSR cloud through TC-ATM interfaces. The ATM edge TSR adds tags to untagged packets and strips tags from tagged packets.

BPX—Broadband Packet Exchange (BPX). The BPX is a carrier quality switch, with trunk and CPU hot standby redundancy.

BXM—Broadband Switch Module. ATM port card for the BPX switch.

VSI—Virtual Switch Interface (VSI). The protocol that enables a TSC to control an ATM switch over an ATM link.

VSI master—In a hardware context, a device that controls a VSI switch (for example, a VSI Tag Switch Controller). In a software context, a process that implements the master side of the VSI protocol.

VSI slave—In a hardware context, a switch or a port card that implements the VSI. In a software context, a process that implements the slave side of the VSI protocol.

extended tag ATM interface—A new type of interface supported by the remote ATM switch driver and a particular switch-specific driver that supports Tag Switching over an ATM interface on a remotely controlled switch.

external ATM interface—One of the interfaces on the controlled ATM switch other than the switch control port. It is also referred to as an exposed ATM interface, because it is available for connections outside of the tag controlled switch.

master control port—A physical interface on a TSC that is connected to one end of a slave control link.

remote ATM switch driver—A set of interfaces that allow IOS software to control the operation of a remote ATM switch through a control protocol, such as VSI.

Ships in the Night (SIN)—The ability to support both Tag Switching procedures and ATM Forum protocols on the same physical interface, or on the same router or switch platform. In this mode, the two protocol stacks operate independently.

controlled ATM switch—An ATM switch that is being controlled by a TSC.

switch control port—An interface that uses a TSC to control the operation of a controlled ATM switch (for example, VSI). The protocol runs on the ATM link.

tag controlled switch—The Tag Switch Controller and controlled ATM switch that it controls, viewed together as a unit.

Tag switch controller (TSC)—An IOS platform that runs the generic Tag Switching software and is capable of controlling the operation of an external ATM (or other type of) switch, making the interfaces of the latter appear externally as TC-ATM interfaces.

tag-switched path (TSP) tunnel—A configured connection between two routers, using Tag Switching to carry the packets.

Tag Switching router (TSR)—A Layer 3 router that forwards packets based on the value of a tag encapsulated in the packets.

TC-ATM interface—A Tag Switching interface where tags are carried in the VPI/VCI bits of ATM cells and where VC connections are established under the control of Tag Switching control software.

TFIB—Tag Forwarding Information Base (TFIB). A data structure and way of managing forwarding in which destinations and incoming tags are associated with outgoing interfaces and tags.

TVC—Tag switched controlled virtual circuit (TVC). A virtual circuit (VC) established under the control of Tag Switching. A TVC is not a PVC or an SVC. It must traverse only a single hop in a tag-switched path (TSP), but may traverse several ATM hops only if it exists within a VP tunnel.

Platforms

The TSC is supported on the following platforms:

- Cisco 7500 series routers—The supported interfaces are the ATM Interface Processor (AIP), Virtual Interface Processor (VIP) and ATM port adapter (PA-A1).
- Cisco 7200 series routers—The supported interface is the ATM port adapter (PA-A1).

Supported MIBs and RFCs

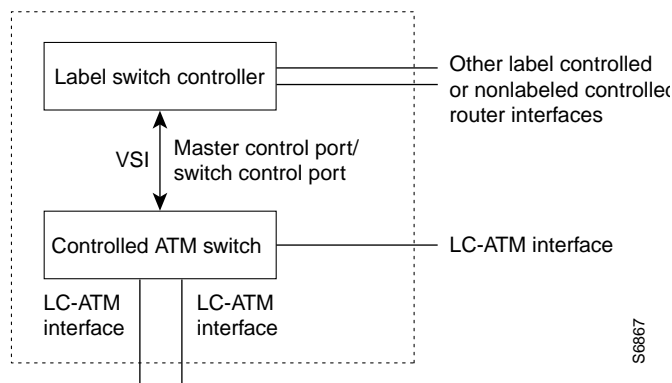
None.

Functional Description

The Tag Switch Controller (TSC) is a tag switch router (TSR) that controls the operation of a separate ATM switch. Together, the router and ATM switch function as a single ATM Tag Switching router (ATM-TSR). A Cisco 7200 or 7500 series router acts as the TSC, and a Cisco BPX 8600 Service Node or a partner's switch acts as the VSI-controlled ATM switch. The TSC controls the ATM switch using the Cisco Virtual Switch Interface (VSI), which runs over an ATM link connecting the two.

The combination of a TSC and the ATM switch it controls is shown in Figure 1.

Figure 1 Tag Switch Controller and Controlled ATM Switch



In this figure, the dotted line represents the external interface of the TSC and controlled switch as seen in the IP routing topology. The controlled ATM switch shows one or more TC-ATM interfaces at this external interface and the TSC itself may have additional interfaces that may or may not be tag controlled.

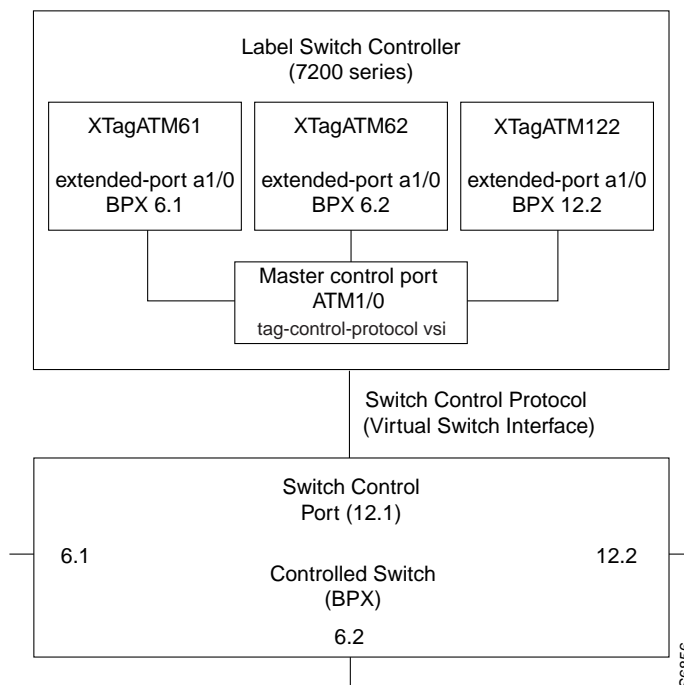
Controlled Switch Ports Represented as Router Interfaces

On the TSC the TC-ATM ports on the controlled switch are represented as a new IOS interface type called extended Tag ATM (XTagATM). XTagATM interfaces are associated with particular physical interfaces on the controlled switch through the **extended-port** interface configuration command.

Figure 2 illustrates a configuration in which a TSC is controlling three ports on a BPX—6.1, 6.2, and 12.2. These corresponding XTagATM interfaces have been created on the TSC and associated with the corresponding ATM ports using the **extended-port** interface configuration command. Note that an additional port on the BPX (12.1) acts as the switch control port and an ATM interface (ATM1/0) on the TSC acts as the master control port.

Figure 2 shows a typical TSC configuration where the TSC and BPX together function as an ATM-TSR.

Figure 2 Typical TSC/BPX Configuration



TSC as Tag Edge Device

The TSC can function simultaneously as a controller for an ATM switch and as a tag edge device. Traffic can be forwarded between a router interface and a TC-ATM interface on the controlled switch as well as between two TC-ATM interfaces on the controlled switch. The TSC can perform the imposition and removal of tags and can serve as the head or tail of a tag-switched path (TSP) tunnel. However, when acting as a tag edge device the TSC is limited by the capabilities of its control link with the switch as follows:

- Total throughput between all other router interfaces and switch interfaces is limited by the bandwidth of the control link (that is, OC-3, 155 Mb per second).
- Tag space for TSC-terminated VCs is limited by the number of VCs supported on the control link.

Support for ATM Forum Protocols

The TSC may be connected to a network running ATM Forum protocols while simultaneously performing its TSC function. However, the connection to the ATM-Forum network must be through a separate ATM interface, that is, not through the master control port.

TSC Configuration Task

This section provides an example of a configuration task for enabling Tag Switching on a Tag Switch Controller (TSC).

Refer to the *Cisco BPX 8600 Series* documentation for the BPX Service Node configuration examples.

Configuring Tag Switching on a TSC-Controlled BPX Port

This case shows how to configure Tag Switching on a port of the BPX that is being controlled by the TSC.

The assumption is that the BPX is connected to the TSC through ATM1/0 and the goal is to configure Tag Switching on slot 6, port 1 of the BPX (see Figure 2).

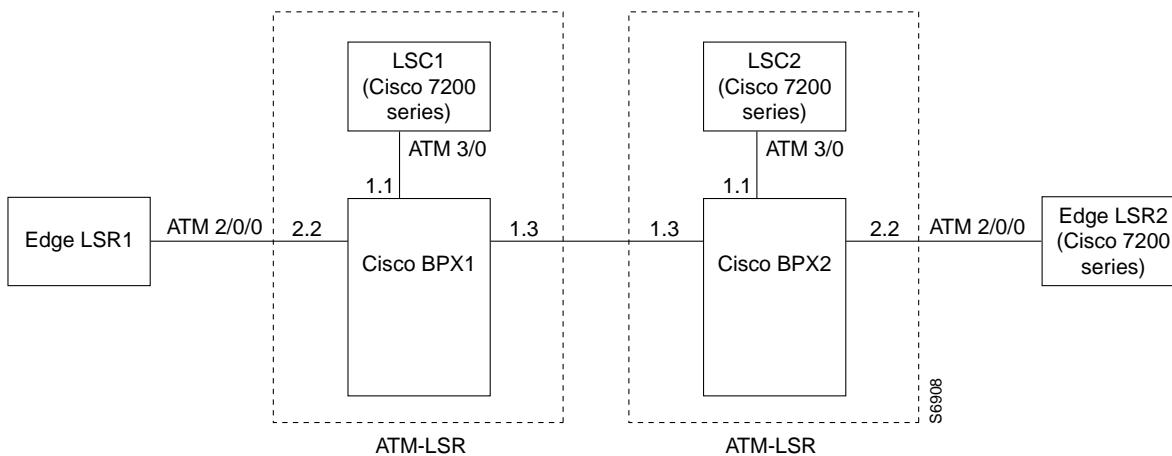
The steps to do this are as follows:

Task	Command
Step 1 Enable the VSI protocol on the control interface (ATM1/0).	Router(config)# interface atm1/0 Router(config-if)# tag-control-protocol vsi
Step 2 Create an extended tag ATM (XTagATM) virtual interface and bind it to BPX port 6.1.	Router(config-if)# interface XTagATM61 Router(config-if)# extended-port atm1/0 bpx 6.1
Step 3 Configure Tag Switching on the extended tag ATM interface. (extended tag ATM interfaces differ from ordinary ATM interfaces in that Tag Switching is configured on the primary interface of an extended tag ATM interface, whereas it is configured on a Tag Switching subinterface of an ordinary ATM interface.)	Router(config-if)# ip address 192.103.210.5 255.255.255.0 Router(config-if)# tag-switching ip Router(config-if)# exit
Step 4 Enable Cisco Express Forwarding (CEF) switching.	Router(config)# ip cef switch

Configuration Example

In this example, the network topology includes ATM-TSRs in a Tag Switching network. It provides configurations for two TSCs (7200 routers), two BPX Service Nodes and two edge TSRs (7500 routers).

Figure 3 ATM-TSR Network Configuration Example



Configuration for TSC1

```

7200 TSC1:
 ip cef switch
 !
 interface ATM3/0
 no ip address
 tag-control-protocol vsi
 !
 interface XTagATM13
 extended-port ATM3/0 bpx 1.3
 !
 ip address 142.4.133.13 255.255.0.0
 tag-switching ip
 !
 interface XTagATM22
 extended-port ATM3/0 bpx 2.2
 !
 ip address 142.6.133.22 255.255.0.0
 tag-switching ip
 !
    
```

Configuration for BPX1 and BPX2

```
BPX1 and BPX2:
  uptrk 1.1
  cnfrsrc 1.1 256 0 1 e 0 2000 1 255 0 353000
  uptrk 1.3
  cnfrsrc 1.3 256 0 1 e 0 2000 1 255 0 353000
  uptrk 2.2
  cnfrsrc 2.2 256 0 1 e 0 2000 1 255 0 353000
  addshelf 1.1 v 1 1
```

Configuration for TSC2

```
7200 TSC2:
  ip cef switch
  !
  interface ATM3/0
  no ip address
  tag-control-protocol vsi slaves 2
  !
  interface XTagATM13
  extended-port ATM3/0 bpx 1.3
  !
  ip address 142.4.143.13 255.255.0.0
  tag-switching ip
  !
  interface XTagATM22
  extended-port ATM3/0 bpx 2.2
  !
  ip address 142.2.143.22 255.255.0.0
  tag-switching ip
  !
```

Configuration for Edge TSR1

```
7500 TSR1:
  ip cef distributed switch
  !
  interface ATM2/0/0
  no ip address
  !
  interface ATM2/0/0.5 tag-switching
  ip address 142.6.132.2 255.255.0.0
  tag-switching ip
  !
```

Configuration for Edge TSR2

```
7500 TSR2:
  ip cef distributed switch
  !
  interface ATM2/0/0
  no ip address
  !
  interface ATM2/0/0.9 tag-switching
  ip address 142.2.142.2 255.255.0.0
  tag-switching ip
  !
```

Command Reference

This section documents new commands. All other commands used with this feature are documented in the Cisco IOS Release 11.2 command references.

- **extended-port**
- **interface XTagATM**
- **show atm vc**
- **show interface XTagATM**
- **show controllers XTagATM**
- **show controllers vsi control-interface**
- **show controllers vsi descriptor**
- **show controllers vsi session**
- **show controllers vsi status**
- **show controllers vsi traffic**
- **show xtagatm cross-connect**
- **show xtagatm vc**
- **tag-control-protocol vsi**
- **tag-switching atm control-vc**
- **tag-switching atm vpi**
- **tag-switching atm vp-tunnel**

extended-port

To associate the currently selected extended tag ATM (XTagATM) interface with a particular external interface on the remotely controlled ATM switch, use the **extended-port** interface configuration command.

```
extended-port ctrl-if {bpx bpx-port-number | descriptor vsi-descriptor | vsi vsi-port-number}
```

Syntax Description

<i>ctrl-if</i>	Identifies the ATM interface used to control the remote ATM switch. VSI must be configured on this interface using the tag-control-protocol interface configuration command.
bpx <i>bpx-port-number</i>	Specifies the associated BPX interface using the native BPX syntax. <i>slot.port</i> [<i>virtual port</i>] Note that this form of the command may only be used when the controlled switch is a BPX.
descriptor <i>vsi-descriptor</i>	Specifies the associated port by its VSI physical descriptor. Note that the <i>vsi-descriptor</i> string must match the corresponding VSI physical descriptor exactly.
vsi <i>vsi-port-number</i>	Specifies the associated port by its VSI logical interface number (integer).

Default

None

Usage Guidelines

The **extended-port** interface configuration command associates an XTagATM interface with a particular external interface on the remotely controlled ATM switch. The three alternate forms of the command permit the external interface on the controlled ATM switch to be specified in three different ways.

Example

The following example shows you how to create an extended tag ATM interface and bind it to the BPX port 2.3.

```
interface XTagATM0
  extended-port atm0/0 bpx 2.3
```

Related Commands

interface XTagATM

interface XTagATM

To enter interface configuration mode for the extended tag ATM (XTagATM) interface, use the **interface XTagATM** command. The interface is created the first time this command is issued for a particular interface number.

```
interface XTagATM if-num
```

Syntax Description

<i>if-num</i>	Interface number.
---------------	-------------------

Default

None

Command Mode

Configuration

Usage Guidelines

Extended tag ATM interfaces are virtual interfaces that are created on first reference like tunnel interfaces. They are similar to ATM interfaces except that they only support TC-ATM encapsulation.

Example

The following example shows you how to create the extended tag ATM interface with the interface number 62:

```
interface XTagATM62
```

Related Commands

extended-port

show atm vc

To display information about private ATM virtual circuits (VCs), use the **show atm vc** privileged EXEC command.

show atm vc [*vcd*]

Private VCs exist on the control interface of a TSC to support corresponding VCs on an extended tag ATM interface.

Syntax Description

<i>vcd</i>	(Optional) Specifies which virtual circuit to display information about.
------------	--

Command Mode

Privileged EXEC

Usage Guidelines

VCs on the extended tag ATM interfaces do not show up in the **show atm vc** command output. Instead, the **show xtagatm vc** command provides a similar output which shows information only on extended tag ATM VCs.

Sample Displays

In the following example, no VCD is specified and private VCs are present.

Router# **show atm vc**

Interface	VCD	VPI	VCI	Type	AAL / Encapsulation	Peak Kbps	Avg. Kbps	Burst Cells	Status
ATM1/0	1	0	40	PVC	AAL5-SNAP	0	0	0	ACTIVE
ATM1/0	2	0	41	PVC	AAL5-SNAP	0	0	0	ACTIVE
ATM1/0	3	0	42	PVC	AAL5-SNAP	0	0	0	ACTIVE
ATM1/0	4	0	43	PVC	AAL5-SNAP	0	0	0	ACTIVE
ATM1/0	5	0	44	PVC	AAL5-SNAP	0	0	0	ACTIVE
ATM1/0	15	1	32	PVC	AAL5-XTAGATM	0	0	0	ACTIVE
ATM1/0	17	1	34	TVC	AAL5-XTAGATM	0	0	0	ACTIVE
ATM1/0	26	1	43	TVC	AAL5-XTAGATM	0	0	0	ACTIVE
ATM1/0	28	1	45	TVC	AAL5-XTAGATM	0	0	0	ACTIVE
ATM1/0	29	1	46	TVC	AAL5-XTAGATM	0	0	0	ACTIVE
ATM1/0	33	1	50	TVC	AAL5-XTAGATM	0	0	0	ACTIVE

Table 1 lists the significant fields in this display.

When you specify a VCD value and the VCD corresponds to that of a private VC on a control interface, the display output appears as follows:

```
Router# show atm vc 15

ATM1/0 33      1      50 TVC AAL5-XTAGATM      0      0      0 ACTIVE
ATM1/0: VCD: 15, VPI: 1, VCI: 32, etype:0x8, AAL5 - XTAGATM, Flags: 0xD38
PeakRate: 0, Average Rate: 0, Burst Cells: 0, VCmode: 0x0
XTagATM1, VCD: 1, VPI: 0, VCI: 32
OAM DISABLED, InARP DISABLED
InPkts: 38811, OutPkts: 38813, InBytes: 2911240, OutBytes: 2968834
InProc: 0, OutProc: 0, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
OAM F5 cells sent: 0, OAM cells received: 0
Status: ACTIVE
```

Table 1 Show ATM VC Command Field Descriptions

Field	Description
ATM1/0	Interface slot and number.
VCD	Virtual circuit descriptor (virtual circuit number).
VPI	Virtual path identifier.
VCI	Virtual circuit identifier.
etype	Ethernet type.
AAL5- XTAGATM	Type of ATM adaptation layer (AAL) and encapsulation. A private VC has AAL5 and encapsulation XTAGATM.
Flags	Bit mask describing virtual circuit information. The flag values are summed to result in the displayed value. 0x10000 ABR VC 0x20000 CES VC 0x40000 TVC 0x100 TEMP (automatically created) 0x200 MULTIPOINT 0x400 DEFAULT_RATE 0x800 DEFAULT_BURST 0x10 ACTIVE 0x20 PVC 0x40 SVC 0x0 AAL5-SNAP 0x1 AAL5-NLPID 0x2 AAL5-FRNLPID 0x3 AAL5-MUX 0x4 AAL3/4-SMDS 0x5 QSAAL 0x6 AAL5-ILMI 0x7 AAL5-LANE 0x8 AAL5-XTAGATM 0x9 CES-AAL1 0xA F4-OAM
PeakRate	Number of packets transmitted at the peak rate.

Field	Description
Average Rate	Number of packets transmitted at the average rate.
Burst Cells	Value that, when multiplied by 32, equals the maximum number of ATM cells the virtual circuit can transmit at the peak rate of the virtual circuit.
VCmode	AIP-specific or NPM-specific register describing the usage of the virtual circuit. Contains values such as rate queue, peak rate, and AAL mode, which are also displayed in other fields.
xtagatm1	Interface of corresponding extended tag ATM VC.
VCD	Virtual circuit descriptor (virtual circuit number) of the corresponding extended tag ATM VC.
VPI	Virtual path identifier of the corresponding extended tag ATM VC.
VCI	Virtual channel identifier of the corresponding extended tag ATM VC.
OAM DISABLED	OAM cells are disabled.
InPkts	Total number of packets received on this virtual circuit. This number includes all silicon-switched, fast-switched, autonomous-switched, and process-switched packets.
OutPkts	Total number of packets sent on this virtual circuit. This number includes all silicon-switched, fast-switched, autonomous-switched, and process-switched packets.
InBytes	Total number of bytes received on this virtual circuit. This number includes all silicon-switched, fast-switched, autonomous-switched, and process-switched packets.
OutBytes	Total number of bytes sent on this virtual circuit. This number includes all silicon-switched, fast-switched, autonomous-switched, and process-switched packets.
InPRoc	Number of process-switched input packets.
OutPRoc	Number of process-switched output packets.
Broadcasts	Number of process-switched broadcast packets.
InFast	Number of fast-switched input packets.
OutFast	Number of fast-switched output packets.
InAS	Number of autonomous-switched or silicon-switched input packets.
OutAS	Number of autonomous-switched or silicon-switched output packets.
OAM F5 cells sent	Number of OAM cells sent on this virtual circuit.
OAM cells received	Number of OAM cells received on this virtual circuit.
Status	Displays the current state of the specified ATM interface.

Table 2 lists the significant fields in this display.

Table 2 Show Interface XTagATM Command Field Descriptions

Field	Description
XTagATM0 is up	Interface is currently active.
line protocol is up	Shows line protocol is up.
Hardware is Tag-Controlled Switch Port	Specifies the hardware type.
Interface is unnumbered	Specifies that this is an unnumbered interface.
MTU	Maximum transmission unit of the extended tag ATM interface.
BW	Bandwidth of the interface in kilobits per second.
DLY	Delay of the interface in microseconds.
rely	Reliability of the interface as a fraction of 255/ (255/255 is 100% reliability), calculated as an exponential average over 5 minutes.
load	Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes.
Encapsulation ATM Tagswitching	Encapsulation method.
loopback not set	Indicates that loopback is not set.
Encapsulation(s)	Identifies the ATM adaptation layer.
Control interface	Identifies the control port switch port with which the extended tag ATM interface has been associated through the extended-port interface configuration command.
9 terminating VCs	Number of terminating VCs with an endpoint on this extended tag ATM interface. Packets are transmitted and/or received by the TSC on a terminating VC, or are forwarded between a TSC-controlled switch port and a router interface.
16 switch cross-connects	Number of switch cross-connects on the external switch with an endpoint on the switch port that corresponds to this interface. This includes cross-connects to terminating VCs that carry data to and from the TSC, as well as cross-connects that bypass the TSC and switch cells directly to other ports.
Switch port traffic:	Number of cells received and transmitted on all cross-connects associated with this interface.
Terminating traffic counts:	Indicates that counters below this line apply only to packets transmitted or received on terminating VCs.
Five minute input rate, Five minute output rate	Average number of bits and packets transmitted per second in the last 5 minutes.
packets input	Total number of error-free packets received by the system.

Field	Description
bytes	Total number of bytes, including data and MAC encapsulation, in the error-free packets received by the system.
no buffer	Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts on noise on serial lines are often responsible for no input buffer events.
broadcasts	Total number of broadcast or multicast packets received by the interface.
runts	Number of packets that are discarded because they are smaller than the medium's minimum packet size.
giants	Number of packets that are discarded because they exceed the medium's maximum packet size.
input errors	Total number of no buffer, runts, giants, CRCs, frame, overrun, ignored and abort counts. Other input-related errors can also increment the count, so that this sum may not balance with the other counts.
CRC	Cyclic redundancy checksum generated by the originating LAN station or far end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRC's is usually the result of collisions or a station transmitting bad data. On a serial link, CRC's usually indicate noise, gain hits or other transmission problems on the data link.
frame	Number of packets received incorrectly having a CRC error and a noninteger number of octets.
overrun	Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data.
ignored	Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different from the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be incremented.
abort	Illegal sequence of one bits on the interface. This usually indicates a clocking problem between the interface and the data link equipment.
packets output	Total number of messages transmitted by the system.
bytes	Total number of bytes, including data and MAC encapsulation, transmitted by the system.

Field	Description
underruns	Number of times that the transmitter has been running faster than the router can handle. This may never be reported on some interfaces.
output errors	Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories.
collisions	Number of messages retransmitted due to an Ethernet collision. This is usually the result of an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). A packet that collides is counted only once in output packets.
interface resets	Number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds. On a serial line, this can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down.
output buffers copied	
interrupts	
failures	

Related Commands

interface XTagATM

Table 3 Show Controllers XTagATM Command Field Descriptions

Field	Description
Interface XTagATM is up	Indicates the overall status of the interface. May be “up”, “down”, or “administratively down.”
Hardware is Tag-Controlled ATM Port	<p>Indicates the hardware type. If the XTagATM has been successfully associated with a switch port, a description of the form "(on <switch_type> switch <name>)" follows this field, where <switch_type> indicates the type of switch (for example, BPX), and "name" is an identifying string learned from the switch.</p> <p>If the XTagATM interface has not been bound to a switch interface (with the extended-port interface configuration command), then the label "Not bound to a control interface and switch port" appears.</p> <p>If the interface has been bound, but the target switch interface has not been discovered by the TSC, then the label "Bound to undiscovered switch port (id <number>)" appears, where <number> is the logical interface ID, in hexadecimal notation.</p>
Control interface ATM1/0 is up	Indicates that the XTagATM interface has been bound (with the extended-port interface configuration command) to the VSI master whose control interface is ATM1/0 and that this control interface is up.
Physical descriptor is ...	The physical descriptor is a string identifying the interface which has been learned from the switch.
Logical interface	This 32-bit quantity, learned from the switch uniquely identifies the interface. It is displayed in both hexadecimal and dotted quad notation
Oper state	The operational state of the interface, according to the switch. One of ACTIVE, FAILED_EXT (that is, in external alarm), FAILED_INT (indicates the inability of the TSC to communicate with the VSI slave controlling the interface, or another internal failure), or REMOVED (administratively removed on the switch).
admin state	The administrative state of the interface, according to the switch—UP or DOWN.
VPI range 1-255	Indicates the allowable VPI range for the interface which has been configured on the switch.
VCI range 32-65535	Indicates the allowable VCI range for the interface which has been configured on, or determined by the switch.
Tag control VC need not be strictly in VPI/VCI range	Indicates that the tag control VC does not need to be within the range specified by VPI range but may be on VPI 0 instead.
Available channels:	Indicates the number of channels (endpoints) which are currently free to be used for cross-connects.
Maximum cell rate:	The maximum cell rate for the interface, which has been configured on the switch.

Field	Description
Available cell rate:	The cell rate which is currently available for new cross-connects on the interface.
Endpoints in use:	The number of endpoints (channels) in use on the interface, broken down by anticipated traffic flow: <ul style="list-style-type: none">• ingress—endpoints carry traffic into the switch• egress— endpoints carry traffic away from the switch• ingress/egress—endpoints carry traffic in both directions.
Rx cells	The number of cells received on the interface.
rx cells discarded	The number of cells received on the interface which were discarded due to traffic management actions. rx header errors.
rx header errors	The number of cells received on the interface with cell header errors.
rx invalid addresses (per card)	The number of cells received with invalid addresses (that is, unexpected VPI/VCI.) On the BPX, this counter is maintained per port group (not per interface.)
last invalid address	The address of the last cell received on the interface with an invalid address (for example, 0/32).
Tx cells	The number of cells transmitted out the interface.
tx cells discarded	The number of cells intended for transmission out the interface that were discarded due to traffic management actions.

Related Commands

show controllers vsi descriptor

show controllers vsi control-interface

To display information about an ATM interface that is configured with the **tag-control-protocol vsi** command to control an external switch, or if an interface is not specified, about all VSI control interfaces, use the **show controllers vsi control-interface** command.

```
show controllers vsi control-interface [interface]
```

Syntax Description

interface (Optional) Interface number.

Command Mode

EXEC

Sample Display

The following is sample output from the **show controllers vsi control-interface** command:

```
Router# show controllers vsi control-interface  
  
Interface:           ATM2/0           Connections:           14
```

The display shows the number of cross-connects currently on the switch that were established by the TSC through VSI over the control interface.

Related Commands

tag-control-protocol vsi

show controllers vsi descriptor

To display information about a switch interface discovered by the TSC through VSI, or if no descriptor is specified, about all such discovered interfaces, use the **show controllers vsi descriptor** command. You specify an interface by its (switch-supplied) physical descriptor.

show controllers vsi descriptor [*descriptor*]

Syntax Description

descriptor (Optional) Physical descriptor. For the BPX, the physical descriptor has the following form:
slot.port.0

Command Mode

EXEC

Usage Guidelines

Per-interface information includes the interface name, the physical descriptor, the interface status, the physical interface state (supplied by the switch), acceptable VPI/VCI ranges, maximum cell rate, available cell rate (forward/backward), and available channels.

Similar information is displayed when you enter the **show controllers XTagATM** command. However, you must specify an IOS interface name instead of a physical descriptor.

Sample Display

The following is sample output from the **show controllers vsi descriptor** command:

```
Router# show controllers vsi descriptor 12.2.0

Phys desc: 12.2.0
Log intf: 0x000C0200 (0.12.2.0)
Interface: XTagATM0
IF status: up                IFC state: ACTIVE
Min VPI: 1                   Maximum cell rate: 10000
Max VPI: 259                 Available channels: 2000
Min VCI: 32                  Available cell rate (forward): 10000
Max VCI: 65535              Available cell rate (backward): 10000
```

Table 4 lists the significant fields in this display.

Table 4 Show Controllers VSI Descriptor Command Field Description

Field	Description
Phys desc	Physical descriptor. A string learned from the switch which identifies the interface.
Log intf	Logical interface ID. This 32-bit quantity, learned from the switch, uniquely identifies the interface.
Interface	The (IOS) interface name.

Field	Description
IF Status	The overall interface status. May be "up", "down", or "administratively down".
Min VPI	Minimum virtual path identifier. Indicates the low end of the VPI range configured on the switch.
Max VPI	Maximum virtual path identifier. Indicates the high end of the VPI range configured on the switch.
Min VCI	Minimum virtual channel identifier. Indicates the low end of the VCI range configured on the switch.
Max VCI	Maximum virtual channel identifier. Indicates the high end of the VCI range configured on, or determined by, the switch.
IFC State	The operational state of the interface, according to the switch. One of FAILED_EXT (that is, in external alarm), FAILED_INT (indicates the inability of the TSC to communicate with the VSI slave controlling the interface, or another internal failure), or REMOVED administratively removed on the switch).
Maximum cell rate	The maximum cell rate for the interface, which has been configured on the switch, in cells per second.
Available channels	Indicates the number of channels (endpoints) which are currently free to be used for cross-connects.
Available cell rate (forward)	The cell rate which is currently available in the forward (that is, ingress) direction for new cross-connects on the interface.
Available cell rate (backward)	The cell rate which is currently available in the backward (that is, egress) direction, for new cross-connects on the interface.

Related Commands

show controllers XTagATM

show controllers vsi session

To display information about all sessions with VSI slaves, use the **show controllers vsi session** command.

Note A session consists of an exchange of VSI messages between the VSI master (the TSC) and a VSI slave (an entity on the switch). There may be multiple VSI slaves for a switch. On the BPX, each port or trunk card assumes the role of a VSI slave.

show controllers vsi session [*session-num* [interface *interface*]]

Syntax Description

session-num Session number.
 interface *interface* VSI control interface.

Command Mode

EXEC

Usage Guidelines

If a session number and an interface are specified, detailed information on the individual session is presented. If the session number is specified but the interface is omitted, detailed information on all sessions with that number is presented. (Only one session can have a given number in the first release, as multiple control interfaces are not supported.)

Sample Display

The following is sample output from the **show controllers vsi session** command:

```
Router# show controllers vsi session
Interface      Session  VCD    VPI/VCI  Switch/Slave Ids  Session State
-----
ATM0/0        0        1      0/40     0/1              ESTABLISHED
ATM0/0        1        2      0/41     0/2              ESTABLISHED
ATM0/0        2        3      0/42     0/3              DISCOVERY
ATM0/0        3        4      0/43     0/4              RESYNC-STARTING
ATM0/0        4        5      0/44     0/5              RESYNC-STOPPING
ATM0/0        5        6      0/45     0/6              RESYNC-UNDERWAY
ATM0/0        6        7      0/46     0/7              UNKNOWN
ATM0/0        7        8      0/47     0/8              UNKNOWN
ATM0/0        8        9      0/48     0/9              CLOSING
ATM0/0        9        10     0/49     0/10             ESTABLISHED
ATM0/0       10        11     0/50     0/11             ESTABLISHED
ATM0/0       11        12     0/51     0/12             ESTABLISHED
```

Table 5 lists the significant fields in this display.

Table 5 Show Controllers VSI Session Command Field Descriptions

Field	Description
Interface	Control interface name.
Session	Session number (from 0 to <n-1>), where n is the number of sessions on the control interface.
VCD	Virtual circuit descriptor (virtual circuit number). Identifies the VC carrying the VSI protocol between the master and the slave for this session.
VPI/VCI	Virtual path identifier/virtual channel identifier (for the VC used for this session.)
Switch/Slave Ids	Switch and slave identifiers supplied by the switch.
Session State	Indicates the status of the session between the master and the slave. ESTABLISHED is the fully operational steady state; UNKNOWN indicates that the slave is not responding. Other possible states include: CONFIGURING RESYNC_STARTING RESYNC_UNDERWAY RESYNC_ENDING DISCOVERY SHUTDOWN_STARTING SHUTDOWN_ENDING INACTIVE

In this example, session number 9 is specified with the **show controllers vsi session** command:

```
Router# show controllers vsi session 9

Interface:          ATM1/0          Session number:      9
VCD:                10              VPI/VCI:            0/49
Switch type:        BPX              Switch id:           0
Controller id:      1                Slave id:            10
Keepalive timer:    15              Powerup session id: 0x0000000A
Cfg/act retry timer: 8/8            Active session id:   0x0000000A
Max retries:        10              Ctrl port log intf: 0x000A0100
Trap window:        50              Max/actual cmd wndw: 21/21
Trap filter:        all              Max checksums:      19
Current VSI version: 1              Min/max VSI version: 1/1
Messages sent:      2502            Inter-slave timer:  4.000
Messages received:  2502            Messages outstanding: 0
```

Table 6 lists the significant fields in this display.

Table 6 Show Controllers VSI Session (With Session Number 9 Specified) Command Field Descriptions

Field	Description
Interface	Name of the control interface on which this session is configured.
Session number	A number from 0 to <n-1>, where n is the number of slaves. Configured on the TSC with the slaves option of the tag-control-protocol vsi command.
VCD	Virtual circuit descriptor (virtual circuit number). Identifies the VC which carries VSI protocol messages for this session.
VPI/VCI	Virtual path identifier/virtual channel identifier, for the VC used for this session.
Switch type	Switch device. For example, the BPX.
Switch id	Switch identifier (supplied by the switch).
Controller id	Controller identifier. Configured on the TSC with the id option of the tag-control-protocol vsi command, and also configured on the switch.
Slave id	Slave identifier (supplied by the switch).
Keepalive timer	VSI master keepalive timeout period, in seconds. Configured on the TSC through the keepalive option of the tag-control-protocol-vsi command. If no valid message is received by the TSC within this period of time, the TSC sends a keepalive message to the slave.
Powerup session id	The session id (supplied by the slave) which it used at powerup time.
cfg/act retry timer	Configured and actual message retry timeout period, in seconds. If no response is received for a command sent by the master within the actual retry timeout period, the message is resent. This applies to most message transmissions. The configured retry timeout value is specified through the retry option of the tag-control-protocol vsi command. The actual retry timeout value is the larger of the configured value and the minimum retry timeout value permitted by the switch.
Active session id	The session ID for the currently active session (supplied by the slave.)
Max retries	The maximum number of times that a particular command transmission will be retried by the master. That is, a message may be sent up to <max_retries+1> times. Configured on the TSC through the retry option of the tag-control-protocol vsi command.
Ctrl port log intf	The logical interface identifier for the control port, as supplied by the switch.
Trap window	The maximum number of outstanding trap messages permitted by the master. This is advertised, but not enforced, by the TSC.

Field	Description
Max/actual cmd wndw	<p>The maximum command window is the maximum number of outstanding (that is., unacknowledged) commands that may be sent by the master before waiting for acknowledgments. This number is communicated to the master by the slave.</p> <p>The actual command window is the maximum number of outstanding commands that are permitted by the master, before it waits for acknowledgments. This is always less than the maximum command window.</p>
Trap filter	This is always “all” for the TSC, indicating that it wishes to receive all traps from the slave. This is communicated to the slave by the master.
Max checksums	The maximum number of checksum blocks supported by the slave. (In this release, the TSC uses only one checksum block.)
Current VSI version	The VSI protocol version currently in use by the master for this session. (In the first release, this is always 1.)
Min/max VSI version	The minimum and maximum VSI versions supported by the slave, as last reported by the slave. If both are zero, the slave has not yet responded to the master.
Messages sent	The number of commands sent to the slave.
Messages received	The number of responses and traps received by the master from the slave for this session.
Messages outstanding	The current number of outstanding messages (that is, commands sent by the master for which responses have not yet been received.)

Related Commands

tag-control-protocol vsi

show controllers vsi status

To display a one-line summary of each VSI-controlled interface, use the **show controllers vsi status** command.

show controllers vsi status

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

If an interface has been discovered by the TSC, but no extended tag ATM interface has been associated with it through the **extended-port** interface configuration command, then the interface name is marked <unknown>, and interface status is marked n/a.

Sample Display

The following is sample output from the **show controllers vsi status** command:

```
Router# show controllers vsi status

Interface Name          IF Status   IFC State   Physical Descriptor
switch control port    n/a        ACTIVE      12.1.0
XTagATM0                up         ACTIVE      12.2.0
XTagATM1                up         ACTIVE      12.3.0
<unknown>              n/a       FAILED-EXT  12.4.0
```

Table 7 lists the significant fields in this display.

Table 7 Show Controllers VSI Status Command Field Descriptions

Field	Description
Interface Name	The (IOS) interface name.
IF Status	The overall interface status. May be "up", "down", or "administratively down."
IFC State	The operational state of the interface, according to the switch. One of FAILED_EXT (that is, in external alarm), FAILED_INT (indicates the inability of the TSC to communicate with the VSI slave controlling the interface, or another internal failure), or REMOVED (administratively removed on the switch).
Physical Descriptor	A string learned from the switch which identifies the interface.

show controllers vsi traffic

To display traffic information about VSI-controlled interfaces, VSI sessions, or VCs on VSI-controlled interfaces, use the **show controllers vsi traffic** command.

```
show controllers vsi traffic [{ descriptor descriptor | session session-num |
vc [descriptor descriptor [vpi vci ]]}]
```

Syntax Description

descriptor <i>descriptor</i>	Specifies the interface.
session <i>session-num</i>	Specifies session number.
<i>vpi</i>	Virtual path identifier.
<i>vci</i>	Virtual circuit identifier.

Command Mode

EXEC

Usage Guidelines

If none of the optional command parameters is specified, traffic for all interfaces is displayed. You can specify a single interface by its (switch-supplied) physical descriptor. For the BPX, the physical descriptor has the form:

```
slot.port. 0
```

If a session number is specified, VSI protocol traffic counts by message type are displayed. The VC traffic display is the same as the one produced by the **show xtagatm vc cross-connect traffic descriptor** command.

Sample Display

The following is sample output from the **show controllers vsi traffic** command:

```
Router# show controllers vsi traffic

Phys desc: 10.1.0
Interface: switch control port
IF status: n/a
Rx cells: 304250           Rx cells discarded: 0
Tx cells: 361186         Tx cells discarded: 0
Rx header errors: 4294967254 Rx invalid addresses (per card): 80360
Last invalid address: 0/53

Phys desc: 10.2.0
Interface: XTagATM0
IF status: up
Rx cells: 202637           Rx cells discarded: 0
Tx cells: 194979         Tx cells discarded: 0
Rx header errors: 4294967258 Rx invalid addresses (per card): 80385
Last invalid address: 0/32

Phys desc: 10.3.0
Interface: XTagATM1
```

```

IF status: up
Rx cells: 182295           Rx cells discarded: 0
Tx cells: 136369         Tx cells discarded: 0
Rx header errors: 4294967262 Rx invalid addresses (per card): 80372
Last invalid address: 0/32
    
```

Table 8 lists the significant fields in this display.

Table 8 Show Controllers VSI Traffic Command Field Descriptions

Field	Description
Phys desc:	The physical descriptor of the interface.
Interface:	The (IOS) interface name.
Rx cells:	The number of cells received on the interface.
Tx cells	The number of cells transmitted on the interface.
Tx cells discarded:	The number of cells which could not be transmitted on the interface due to traffic management and which were therefore discarded.
Rx header errors:	The number of cells which were discarded due to ATM header errors.
Rx cells discarded:	The number of cells received on the interface which were discarded due to traffic management.
Rx invalid addresses:	The number of cells received with an invalid address (that is, an unexpected VPI/VCI combination). With the BPX, this count is of all such cells received on all interfaces in the port group of this interface.
Last invalid address:	The number of cells received on this interface with ATM cell header errors.

The following sample output is displayed when you enter the **show controllers vsi traffic session 9** command:

```

Router# show controllers vsi traffic session 9
                                     Sent                               Received
Sw Get Cnfg Cmd:                    3656           Sw Get Cnfg Rsp:           3656
Sw Cnfg Trap Rsp:                   0              Sw Cnfg Trap:              0
Sw Set Cnfg Cmd:                     1              Sw Set Cnfg Rsp:           1
Sw Start Resync Cmd:                 1              Sw Start Resync Rsp:       1
Sw End Resync Cmd:                   1              Sw End Resync Rsp:         1
Ifc Getmore Cnfg Cmd:                1              Ifc Getmore Cnfg Rsp:      1
Ifc Cnfg Trap Rsp:                   4              Ifc Cnfg Trap:             4
Ifc Get Stats Cmd:                   8              Ifc Get Stats Rsp:         8
Conn Cmt Cmd:                        73             Conn Cmt Rsp:              73
Conn Del Cmd:                        50             Conn Del Rsp:              0
Conn Get Stats Cmd:                  0              Conn Get Stats Rsp:        0
Conn Cnfg Trap Rsp:                  0              Conn Cnfg Trap:            0
Conn Bulk Clr Stats Cmd:              0              Conn Bulk Clr Stats Rsp:   0
Gen Err Rsp:                         0              Gen Err Rsp:               0
unused:                              0              unused:                     0
unknown:                             0              unknown:                    0
TOTAL:                               3795           TOTAL:                       3795
    
```

Table 9 lists the significant fields in this display.

Table 9 Show Controllers VSI Traffic Session Command Field Descriptions

Field	Description
Sw Get Cnfg Cmd	Number of VSI "get switch configuration command" messages sent.
Sw Cnfg Trap Rsp	Number of VSI switch configuration asynchronous trap response messages sent.
Sw Set Cnfg Cmd	Number of VSI "set switch configuration command" messages sent.
Sw Start Resync Cmd	Number of VSI "set resynchronization start command" messages sent.
Sw End Resync Cmd	Number of VSI "set resynchronization end command" messages sent.
Ifc Getmore Cnfg Cmd	Number of VSI "get more interfaces configuration command" messages sent.
Ifc Cnfg Trap Rsp	Number of VSI "interface configuration asynchronous trap response" messages sent.
Ifc Get Stats Cmd	Number of VSI "get interface statistics command" messages sent.
Conn Cmt Cmd	Number of VSI "set connection committed command" messages sent.
Conn Del Cmd	Number of VSI "delete connection command" messages sent.
Conn Get Stats Cmd	Number of VSI "get connection statistics command" messages sent.
Conn Cnfg Trap Rsp	Number of VSI "connection configuration asynchronous trap response" messages sent.
Conn Bulk Clr Stats Cmd	Number of VSI "bulk clear connection statistics command" messages sent.
Gen Err Rsp	Number of VSI "generic error response" messages sent or received.
Sw Get Cnfg Rsp	Number of VSI "get connection configuration command response" messages received.
Sw Cnfg Trap	Number of VSI "switch configuration asynchronous trap" messages received.
Sw Set Cnfg Rsp	Number of VSI "set switch configuration response" messages received.
Sw Start Resync Rsp	Number of VSI "set resynchronization start response" messages received.
Sw End Resync Rsp	Number of VSI "set resynchronization end response" messages received.
Ifc Getmore Cnfg Rsp	Number of VSI "get more interfaces configuration response" messages received.
Ifc Cnfg Trap	Number of VSI "interface configuration asynchronous trap" messages received.

Command Reference

Field	Description
Ifc Get Stats Rsp	Number of VSI "get interface statistics response" messages received.
Conn Cmt Rsp	Number of VSI "set connection committed response" messages received.
Conn Del Rsp	Number of VSI "delete connection response" messages received.
Conn Get Stats Rsp	Number of VSI "get connection statistics response" messages received.
Conn Cnfg Trap	Number of VSI "connection configuration asynchronous trap" messages received.
Conn Bulk Clr Stats Rsp	Number of VSI "bulk clear connection statistics response" messages received.
TOTAL	Total number of VSI messages sent or received.

show xtagatm cross-connect

To display information about the TSC's view of the cross-connect table on the remotely controlled ATM switch, use the **show xtagatm cross-connect** command.

```
show xtagatm cross-connect [traffic] [{interface interface [vpi vci] |  
descriptor descriptor [vpi vci]]
```

Syntax Description

traffic	Displays receive and transmit cell counts for each connection.
interface <i>interface</i>	Displays only connections with an endpoint of the specified interface.
<i>vpi vci</i>	Displays only detailed information on the endpoint with the specified VPI/VCI on the specified interface.
descriptor <i>descriptor</i>	Displays only connections with an endpoint on the interface with the specified physical descriptor.

Command Mode

EXEC

Usage Guidelines

Sample Displays

Each connection is listed twice in the sample output from the **show xtagatm vc cross-connect** command under each interface that is linked by the connection. Connections are marked as “->” (unidirectional traffic flow, into the first interface), “<-” (unidirectional traffic flow, away from the interface) or “<->” (bidirectional).

The following is sample output from the **show xtagatm cross-connect** command:

```
Router# show xtagatm cross-connect
```

Phys Desc	VPI/VCI	Type	X-Phys Desc	X-VPI/VCI	State
10.1.0	1/37	->	10.3.0	1/35	UP
10.1.0	1/34	->	10.3.0	1/33	UP
10.1.0	1/33	<->	10.2.0	0/32	UP
10.1.0	1/32	<->	10.3.0	0/32	UP
10.1.0	1/35	<-	10.3.0	1/34	UP
10.2.0	1/57	->	10.3.0	1/49	UP
10.2.0	1/53	->	10.3.0	1/47	UP
10.2.0	1/48	<-	10.1.0	1/50	UP
10.2.0	0/32	<->	10.1.0	1/33	UP
10.3.0	1/34	->	10.1.0	1/35	UP
10.3.0	1/49	<-	10.2.0	1/57	UP
10.3.0	1/47	<-	10.2.0	1/53	UP
10.3.0	1/37	<-	10.1.0	1/38	UP
10.3.0	1/35	<-	10.1.0	1/37	UP

```

10.3.0      1/33      <-      10.1.0      1/34      UP
10.3.0      0/32      <->     10.1.0      1/32      UP

```

Table 10 lists the significant fields in this display.

Table 10 Show xtagatm Cross-Connect Command Field Descriptions

Field	Description
Phys desc	Physical descriptor. A switch-supplied string identifying the interface on which the endpoint exists.
VPI/VCI	The virtual path identifier and virtual channel identifier for this endpoint.
Type	“->” indicates an ingress endpoint, where traffic is only expected to be received into the switch; “<-” indicates an egress endpoint, where traffic is only expected to be transmitted out the interface; “<->” indicates that traffic is expected to be both transmitted and received at this endpoint.
X-Phys desc	The physical descriptor for the interface of the other endpoint belonging to the cross-connect.
X-VPI/VCI	The virtual path identifier and virtual channel identifier of the other endpoint belonging to the cross-connect.
State	Indicates the status of the cross-connect to which this endpoint belongs. Is typically “UP”; other values, all of which should be transient, include: DOWN ABOUT_TO_DOWN ABOUT_TO_CONNECT CONNECTING ABOUT_TO_RECONNECT RECONNECTING ABOUT_TO_RESYNC RESYNCING NEED_RESYNC_RETRY ABOUT_TO_RESYNC_RETRY RETRYING_RESYNC ABOUT_TO_DISCONNECT DISCONNECTING

A sample of the detailed information provided for a single endpoint is:

```

Router# show xtagatm cross-connect descriptor 12.1.0 1 42

Phys desc: 12.1.0
Interface: n/a
Intf type: switch control port
VPI/VCI: 1/42
X-Phys desc: 12.2.0
X-Interface: XTagATM0

```

```

X-Intf type: extended tag ATM
X-VPI/VCI: 2/38
Conn-state: UP
Conn-type: input/output
Cast-type: point-to-point
Rx service type: Tag COS 0
Rx cell rate: n/a
Rx peak cell rate: 10000
Tx service type: Tag COS 0
Tx cell rate: n/a
Tx peak cell rate: 10000
    
```

Table 11 lists the significant fields in this display.

Table 11 Show xtagatm Cross-Connect Descriptor Field Descriptions

Field	Description
Phys desc	Physical descriptor. A switch-supplied string identifying the interface on which the endpoint exists.
Interface	The (IOS) interface name.
Intf type	Interface type. Either “extended tag ATM” or “switch control port”.
VPI/VCI	The virtual path identifier and virtual channel identifier for this endpoint.
X-Phys desc	The physical descriptor for the interface of the other endpoint belonging to the cross-connect.
X-Interface	The (IOS) name for the interface of the other endpoint belonging to the cross-connect.
X-Intf type	The interface type for the interface of the other endpoint belonging to the cross-connect.
X-VPI/VCI	The virtual path identifier and virtual channel identifier of the other endpoint belonging to the cross-connect.
Conn-state	Indicates the status of the cross-connect to which this endpoint belongs. Is typically “UP”; other values, all of which should be transient, include DOWN ABOUT_TO_DOWN ABOUT_TO_CONNECT CONNECTING ABOUT_TO_RECONNECT RECONNECTING ABOUT_TO_RESYNC RESYNCING NEED_RESYNC_RETRY ABOUT_TO_RESYNC_RETRY RETRYING_RESYNC ABOUT_TO_DISCONNECT DISCONNECTING.

Field	Description
Conn-type	“input” indicates an ingress endpoint where traffic is only expected to be received into the switch; “output” indicates an egress endpoint, where traffic is only expected to be transmitted out the interface; “input/output” indicates that traffic is expected to be both transmitted and received at this endpoint.
Cast-type	Indicates whether or not the cross-connect is multicast. In the first release, this is always point-to-point.
Rx service type	Class of service type for the receive, or ingress, direction. This will be “Tag COS <n>,” (Tag Class of Service <n>), where n is in the range 0-7, for input and input/output endpoints; this will be “n/a” for output endpoints. (In the first release, this is either 0 or 7).
Rx cell rate	(guaranteed) cell rate in the receive, or ingress, direction. In the first release, this is always n/a.
Rx peak cell rate	Peak cell rate in the receive, or ingress, direction, in cells per second. This is “n/a” for an output endpoint.
Tx service type	Class of service type for the transmit, or egress, direction. This will be “Tag COS <n>,” (Tag Class of Service <n>), where n is in the range 0-7, for output and input/output endpoints; this will be “n/a” for input endpoints. (In the first release, n will be either 0 or 7).
Tx cell rate	(guaranteed) cell rate in the transmit, or egress, direction. In the first release, this is always “n/a.”
Tx peak cell rate	Peak cell rate in the transmit, or egress, direction, in cells per second. This is “n/a” for an input endpoint.

show xtagatm vc

To display information about terminating VCs on extended tag ATM (XTagATM) interfaces, use the **show xtagatm vc** command.

```
show xtagatm vc [vcd [interface]]
```

Syntax Description

<i>vcd</i>	(Optional) Virtual circuit descriptor (virtual circuit number). If you specify the <i>vcd</i> argument, then detailed information about all VCs with that <i>vcd</i> is displayed. If you do not specify the <i>vcd</i> argument, then a summary description of all VCs on all XTagATM interfaces is displayed.
<i>interface</i>	(Optional) Interface number. If you specify the <i>interface</i> and the <i>vcd</i> arguments, then the single VC with the specified <i>vcd</i> on the specified <i>interface</i> is selected

Each connection is listed twice in the sample output from the **show xtagatm vc cross-connect** command under each interface that is linked by the connection. Connections are marked as input (unidirectional traffic flow, into the interface), output (unidirectional traffic flow, away from the interface) or in/out (bidirectional).

Command Mode

EXEC

Usage Guidelines

The columns marked VCD, VPI and VCI display information for the corresponding private VC on the control interface. The private VC connects the XTagATM VC to the external switch. It is termed private because its VPI and VCI are only used for communication between the TSC and the switch, and is different from the VPI and VCI seen on the XTagATM interface and the corresponding switch port.

Sample Display

The following is sample output from the **show xtagatm vc** command:

```
Router# show xtagatm vc
```

Interface	VCD	VPI	VCI	Type	Encapsulation	AAL / Control Interface			
						VCD	VPI	VCI	Status
XTagATM0	1	0	32	PVC	AAL5-SNAP	2	0	33	ACTIVE
XTagATM0	2	1	33	TVC	AAL5-MUX	4	0	37	ACTIVE
XTagATM0	3	1	34	TVC	AAL5-MUX	6	0	39	ACTIVE

Table 12 lists the significant fields in this display.

Table 12 Show xtagatm vc Command Field Descriptions

Field	Description
VCD	Virtual circuit descriptor (virtual circuit number).
VPI	Virtual path identifier.
VCI	Virtual circuit identifier.
Control Interf. VCD	VCD for the corresponding private VC on the control interface.
Control Interf. VPI	VPI for the corresponding private VC on the control interface.
Control Interf. VCI	VCI for the corresponding private VC on the control interface.
Encapsulation	Displays the type of connection on the interface.
Status	Displays the current state of the specified ATM interface.

Related Commands

- show atm vc**
- show xtagatm cross-connect**

tag-control-protocol vsi

To configure the use of VSI on a particular master control port, use the **tag-control-protocol vsi** command. To disable VSI, use the **no** form of this command.

```
[no] tag-control-protocol vsi [id controller-id] [base-vc vpi vci] [slaves slave-count]
[keepalive timeout] [retry timeout count]
```

Syntax Description

id <i>controller-id</i>	Determines the value of the controller-id field present in the header of each VSI message. The default is 1.
base-vc <i>vpi vci</i>	Determines the VPI/VCI value for the channel to the first slave. Together with the slaves value, this determines the VPI/VCI values for the channels to all the slaves, which are <i>vpi/vci</i> <i>vpi/vci+1, and so on.</i> <i>vpi/vci+slave_count-1.</i> The default is 0/40.
slaves <i>slave-count</i>	Determines the number of slaves reachable through this master control port. The default is 14 (suitable for the BPX). In the first release, at most twelve sessions will be established with the BPX. The default of 14 will attempt sessions with cards 7 and 8, but such sessions are not used in this release, and is always marked as UNKNOWN.
keepalive <i>timeout</i>	Determines the value of the keepalive timer (in seconds). Note that the keepalive timer value should be greater than the value of the <i>retry_timer</i> times the <i>retry_count+1</i> . The default is 15 seconds.
retry <i>timeout count</i>	Determines the value of the message retry timer (in seconds) and the maximum number of retries. The default is 8 seconds, 10 retries.

Command Mode

Interface configuration

Usage Guidelines

The command is only available on interfaces that can serve as a VSI master control port. It is recommended that all options to the **tag-control-protocol** command be entered at once.

Once VSI is active on the control interface (through an earlier **tag-control-protocol vsi** command), re-entering the command may cause all associated XTagATM interfaces to go down and come back up. In particular, re-entering the **tag-control-protocol vsi** command with any of the following options causes VSI to be shut down and re-activated on the control interface:

- **id**
- **base-vc**
- **slaves**

VSI remains continuously active (that is, will not be shut down and re-activated) if **tag-control-protocol vsi** command is re-entered with only one or more of the following options:

- **keepalive**
- **retry**

In either case, re-entering the **tag-control-protocol vsi** command causes the specified options to take on the newly specified values; the other options retain their previous values. To restore default values to all the options, enter the **no tag-control-protocol** command, followed by the **tag-control-protocol vsi** command.

Example

The following example shows you how to configure the VSI driver on the control interface:

```
interface atm 0/0
tag-control-protocol vsi 0 51
```

tag-switching atm control-vc

To configure the VPI and VCI values to be used for the initial link to the Tag Switching peer, use the **tag-switching atm control-vc** interface configuration command. This link is used to establish the TDP session and to carry non-IP traffic.

tag-switching atm control-vc *vpi vci*

Syntax Description

<i>vpi</i>	Virtual path identifier.
<i>vci</i>	Virtual circuit identifier.

Command Mode

Interface configuration

Usage Guidelines

On an extended tag ATM (XTagATM) interface, the default VPI range to be used for tagged VCs is the configured VPI range that is learned from the switch. This default range should be sufficient for most applications. Use the **tag-switching vpi** command on an XTagATM interface only when it is necessary to override these defaults.

For the **tag-switching atm vpi** command, the VPI range specified must lie within the range that was configured on the BPX for the corresponding BPX interface.

Example

The following example shows you how to create a Tag Switching subinterface on a router and how to select VPI 1 and VCI 34 as the control VC.

```
interface atm4/0.1 tag-switching
tag-switching ip
tag-switching atm control-vc 1 34
```

tag-switching atm vpi

To configure the range of values to use in the VPI field for tag VCs, use the **tag-switching atm vpi** command. To clear the interface configuration, use the **no** form of this command.

```
tag-switching atm vpi vpi [- vpi]
no tag-switching atm vpi vpi [- vpi]
```

Syntax Description

<i>vpi</i>	Virtual path identifier (low end of range).
<i>-vpi</i>	(Optional.) Virtual path identifier (high end of range).

Default

The default is 1-1.

Command Mode

Interface configuration

Usage Guidelines

To configure ATM Tag Switching on a router interface (for example, an ATM Interface Processor), you must enable a Tag Switching subinterface.

Note The **tag-switching atm control-vc** and **tag-switching atm vpi** subinterface level configuration commands are available on any interface that can support ATM tagging.

Use this command to select an alternate range of VPI values for ATM tag assignment on this interface. The two ends of the link negotiate a range defined by the intersection of the range configured at each end.

Example

The following example shows you how to create a subinterface and how to select a VPI range from VPI 1 to VPI 3:

```
interface atm4/0.1 tag-switching
tag-switching ip
tag-switching atm vpi 1-3
```

Related Commands

tag-switching atm control-vc

tag-switching atm vp-tunnel

To specify an interface or a subinterface as a VP tunnel, use the **tag-switching atm vp-tunnel** command.

```
tag-switching atm vp-tunnel vpi
```

Syntax Description

<i>vpi</i>	Provides the VPI value for the local end of the tunnel.
------------	---

Command Mode

Interface configuration

Usage Guidelines

The **tag-switching atm vp-tunnel** and **tag-switching atm vpi** commands are mutually exclusive.

This command is available on both extended tag ATM interfaces and on TC-ATM subinterfaces of ordinary router ATM interfaces. The command is not available on the 1010, where all subinterfaces are automatically VP tunnels.

On an XTagATM interface, the tunnel/non-tunnel status and the VPI value to be used in case the XTagATM interface is a tunnel, are normally learned from the switch through VSI interface discovery. Therefore, it should not be necessary to use the **tag-switching atm vp-tunnel** command on an XTagATM interface in most applications.

Example

The following example shows you how to specify a Tag Switching subinterface VP tunnel, with a VPI value 4.

```
tag-switching atm vp-tunnel 4
```

Debug Commands

This section contains an alphabetical listing of the TSC debug commands and their descriptions. Documentation for each command includes a brief description of its use, command syntax, usage guidelines, sample output, and a description of that output.

- **debug tag-switching xtagatm cross-connect**
- **debug tag-switching xtagatm events**
- **debug tag-switching xtagatm vc**
- **debug tag-switching xtagatm errors**
- **debug vsi api**
- **debug vsi errors**
- **debug vsi events**
- **debug vsi packets**
- **debug vsi param-groups**

debug tag-switching xtagatm cross-connect

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

[no] debug tag-switching xtagatm cross-connect

Usage Guidelines

You can 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 every cross-connect operation performed by the Tag Switch Controller. In a network configuration with a large number of tagged virtual circuits (TVCs) the volume of output generated may interfere with system timing and the correct operation of other router functions. Use this command only in situations where the TVC setup or teardown rate is low.

Sample Display

The following is an example of the display you see when you enter **debug tag-switching xtagatm cross-connect**:

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

XTagATM: cross-conn request; SETUP, userdata 0x17, userbits 0x1, prec 7
          0xC0100 (Ctl-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 13 lists the significant fields shown in this display.

Table 13 Debug Tag-Switching xtagatm Cross-Connect Command Field Description

Field	Description
XTagATM	Identifies the source of the debug message as an XTagATM interface.
cross-conn	Indicates that the debug message pertains to cross-connect setup or teardown operation.
request	Request from an XTagATM interface to the VSI master to setup or teardown a cross-connect.
response	Response from the VSI master to an XTagATM interface that a cross-connect has been setup or removed.
SETUP	The request is for the setup of a cross-connect.
TEARDOWN	The request is 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 which is returned in the corresponding fields 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 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 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

show xtagatm cross-connect

debug tag-switching xtagatm events

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

[no] debug tag-switching xtagatm events

Usage Guidelines

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

- 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 goes down to ensure that they are cleaned up before new XTagATM VCs can be created on the interface.

Sample Display

The following is an example of the display you see when you enter **debug tag-switching xtagatm events**:

```
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 14 lists the significant fields shown in this display.

Table 14 Debug Tag-Switching xtagatm Events Command Field Description

Field	Description
XTagATM	Identifies the source of the debug message as the XTagATM 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.

Field	Description
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 extended-port interface configuration command) to any XTagATM interface.
marking all VCs stale on XTagATM0	Indicates that all existing XTagATM VCs on XTagATM0 have been marked as stale, and that XTagATM0 remains down until all of these VCs have been cleaned up.

debug tag-switching xtagatm vc

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

[no] debug tag-switching xtagatm vc

Usage Guidelines

You can 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 the number of XTagATM VCs being setup or torn down is large, and this output may interfere with system timing and normal operation of other router functions. Use the **debug tag-switching xtagatm vc** command only in situations where the number of XTagATM VCs being created or removed is small.

Sample Display

The following is an example of the display you see when you enter **debug**:

```
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 15 lists the significant fields shown in this display.

Table 15 Debug Tag-Switching xtagatm VC Command Field Description

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 for the terminating VC.
vcd vpi/vci	Indicates the VCD and VPI and VCI value for the terminating VC.
(ctl-if vcd vpi/vci)	Shows the control interface, and the VCD and VPI and VCI value for the private VC corresponding the to XTagATM vc on the control interface.
Setup, Complete, Teardown	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 the indicated event.

debug tag-switching xtagatm errors

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

[no] debug tag-switching xtagatm errors

Usage Guidelines

You can use the **debug tag-switching xtagatm errors** command to display information about abnormal conditions and events that occur on XTagATM interfaces.

Sample Display

The following is an example of the display you see when you enter **debug tag-switching xtagatm errors**:

```
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 the a terminating VC on XTagATM0 failed, and that the reason for the failure was a lack of resources on the controlled ATM switch.

debug vsi api

Use the **debug vsi api** EXEC command to display information on events associated with the external ATM API interface to the VSI master. The **no** form of this command disables debugging output.

[no] debug vsi api

Usage Guidelines

You can use **debug vsi api** command to monitor the communication between the VSI master and the XTagATM component about interface changes and cross-connect requests.

Sample Displays

The following is an example of the display you see when you enter **debug vsi api**:

```
Router# debug vsi api

VSI_M: vsi_exatm_conn_req: 0x000C0200/1/35 -> 0x000C0100/1/50
        desired state up, status OK
VSI_M: vsi_exatm_conn_resp: 0x000C0200/1/33 -> 0x000C0100/1/49
        curr state up, status OK
```

Table 16 lists the significant fields shown in this display.

Table 16 Debug VSI API Command Field Description

Field	Description
vsi_exatm_conn_req	Indicates that a connect or disconnect request has been submitted to the VSI Master.
0x000C0200	The logical interface identifier of the primary endpoint, in hexadecimal form.
1/35	The VPI/VCI of the primary endpoint.
->	Indicates that the expected traffic flow is unidirectional (from the primary endpoint to the secondary endpoint). The other value for this field is "<->," which indicates bidirectional traffic flow.
0x000C0100	The logical interface identifier of the secondary endpoint.
1/50	The VPI/VCI of the secondary endpoint.
desired state	"up" indicates a connect request, while "down" indicates a disconnect request.
status (in vsi_exatm_conn_req output)	A mnemonic indicating the success or failure of the initial processing of the request. One of OK, INVALID_ARGS, NONEXIST_INTF, TIMEOUT, NO_RESOURCES, FAIL. Note that OK means only that the request has been successfully queued for transmission to the switch; it does not indicate completion of the request.

debug vsi errors

Use the **debug vsi errors** EXEC command to display information on errors encountered by the VSI Master. The **no** form of this command disables debugging output.

[no] debug vsi errors [interface *interface* [slave *number*]]

Syntax Description

interface <i>interface</i>	Interface number.
slave <i>number</i>	Slave number (beginning with zero).

Usage Guidelines

You can use the **debug vsi errors** command to display information on errors encountered by the VSI master when parsing received messages, as well as information on unexpected conditions encountered by the VSI Master.

If the interface parameter is specified, output is restricted to errors associated with the indicated VSI control interface. If the slave number is specified, output is further restricted to errors associated with the session with the indicated slave.

Note Slave numbers are the same as the session numbers discussed under the **show controllers vsi session** command.

Multiple uses of the form of the command which specifies slave number allows multiple slaves to be debugged at once. For example, the following commands restrict output to that for errors associated with sessions 0 and 1 on control interface atm2/0 (but for no other sessions).

```
Router# debug vsi errors interface atm2/0 slave 0
Router# debug vsi errors interface atm2/0 slave 1
```

Some errors are not associated with any particular control interface or session; messages associated with these errors are printed regardless of the **interface** or **slave** options which are in effect.

Sample Display

The following is an example of the display you see when you enter **debug vsi errors**:

```
Router# debug vsi errors

VSI Master: parse error (unexpected param-group contents) in GEN ERROR RSP rcvd on
ATM2/0:0/51 (slave 0)
          errored section is at offset 16, for 2 bytes:
          01.01.00.a0 00.00.00.00 00.12.00.38 00.10.00.34
          *00.01*00.69 00.2c.00.00 01.01.00.80 00.00.00.08
          00.00.00.00 00.00.00.00 00.00.00.00 0f.a2.00.0a
          00.01.00.00 00.00.00.00 00.00.00.00 00.00.00.00
          00.00.00.00
```

Table 17 lists the significant fields shown in this display.

Table 17 Debug VSI Errors Command Field Description

Field	Description
parse error	Indicates that an error has been encountered while parsing a message received by the VSI master.
unexpected param-group contents	Indicates the type of parsing error encountered. In this case, a parameter group within the message contained invalid data.
GEN ERROR RSP	A mnemonic for the function code in the header of the errored message.
ATM2/0	The control interface on which the errored message was received.
0/51	The VPI/VCI of the VC (on the control interface) on which the errored message was received.
slave	The number of the session on which the errored message was received.
offset <n>	Indicates the number of bytes between the start of the VSI header the start of the errored portion of the message.
<n> bytes	The length of the errored section.
00.01.00.a0 [...]	The entire errored message, as a series of hexadecimal bytes. Note that the errored section is between asterisks (*).

debug vsi events

Use the **debug vsi events** EXEC command to display information on events that affect entire sessions as well as events that affect only individual connections. The **no** form of this command disables debugging output.

```
[no] debug vsi events [interface interface [slave number]]
```

Syntax Description

interface <i>interface</i>	Interface number.
slave <i>number</i>	Slave number (beginning with zero).

Usage Guidelines

You can use the **debug vsi events** command to display information on events associated with the per-session state machines of the VSI master, as well as the per-connection state machines. If the interface parameter is specified, output is restricted to events associated with the indicated VSI control interface. If the slave number is specified, output is further restricted to events associated with the session with the indicated slave.

Note Slave numbers are the same as the session numbers discussed under the **show controllers vsi session** command.

Multiple uses of the form of the command which specifies slave number allows multiple slaves to be debugged at once. For example, the following commands restrict output to that for events associated with sessions 0 and 1 on control interface atm2/0 (but for no other sessions). Output associated with all per-connection events are displayed regardless of the **interface** or **slave** options which are in effect.

```
Router# debug vsi events interface atm2/0 slave 0
Router# debug vsi events interface atm2/0 slave 1
```

Sample Display

The following is an example of the display you see when you enter **debug vsi events**:

```
Router# debug vsi events

VSI Master: conn 0xC0200/1/37->0xC0100/1/51:
CONNECTING -> UP
VSI Master(session 0 on ATM2/0):
event CONN_CMT_RSP, state ESTABLISHED -> ESTABLISHED
VSI Master(session 0 on ATM2/0):
event KEEPALIVE_TIMEOUT, state ESTABLISHED -> ESTABLISHED
VSI Master(session 0 on ATM2/0):
event SW_GET_CNFG_RSP, state ESTABLISHED -> ESTABLISHED
debug vsi packets
```

Table 18 lists the significant fields shown in this display.

Table 18 Debug VSI Events Command Field Description

Field	Description
conn	Indicates that the event applies to a particular connection.
0xC0200	The logical interface identifier of the primary endpoint, in hexadecimal form.
1/37	The VPI/VCI of the primary endpoint.
->	Indicates the expected traffic flow is unidirectional (from the primary endpoint to the secondary endpoint.) The other value for this field is "<->," indicating bidirectional traffic flow.
0xC0100	The logical interface identifier of the secondary endpoint.
1/51	The VPI/VCI of the secondary endpoint.
<state1> -> <state2>	<state1> is a mnemonic for the state of the connection before the event occurred; <state2> represents the state of the connection after the event occurred.
session	Indicates the number of the session with which the event is associated.
ATM2/0	Indicates the control interface associated with the session.
event	A mnemonic for the event that has occurred. This includes mnemonics for the function codes of received messages (for example, CONN_CMT_RSP), as well as mnemonics for other sorts of events (for example, KEEPALIVE_TIMEOUT).
state <state1> -> <state2>	Mnemonics for the session states associated with the transition triggered by the event. <state1> is a mnemonic for the state of the session before the event occurred; <state2> is a mnemonic for the state of the session after the event occurred.

debug vsi packets

Use the **debug vsi packets** EXEC command to display a one-line summary of each VSI message sent and received by the TSC. The **no** form of this command disables debugging output.

[no] debug vsi packets [interface *interface* [slave *number*]]

Syntax Description

interface <i>interface</i>	Interface number.
slave <i>number</i>	Slave number (beginning with zero).

Usage Guidelines

If the interface parameter is specified, output is restricted to messages sent and received on the indicated VSI control interface. If the slave number is specified, output is further restricted to messages sent and received on the session with the indicated slave.

Note Slave numbers are the same as the session numbers discussed under the **show controllers vsi session** command.

Multiple uses of the form of the command which specifies slave number allows multiple slaves to be debugged at once. For example, the following commands restrict output to that for messages received on atm2/0 for sessions 0 and 1, (but for no other sessions).

```
Router# debug vsi packets interface atm2/0 slave 0
Router# debug vsi packets interface atm2/0 slave 1
```

Sample Display

The following is an example of the display you see when you enter **debug vsi packets**:

```
Router# debug vsi packets

VSI master(session 0 on ATM2/0): sent msg SW GET CNFG CMD on 0/51
VSI master(session 0 on ATM2/0): rcvd msg SW GET CNFG RSP on 0/51
VSI master(session 0 on ATM2/0): sent msg SW GET CNFG CMD on 0/51
VSI master(session 0 on ATM2/0): rcvd msg SW GET CNFG RSP on 0/51
```

Table 19 lists the significant fields shown in this display.

Table 19 **Debug VSI Packets Command Field Description**

Field	Description
session	Session number identifying a particular VSI slave. Numbers begin with zero. See the show controllers vsi session command.
ATM2/0	The identifier for the control interface on which the message was sent or received.
sent	The message was sent by the VSI master.
rcvd	The message was received by the VSI master.
msg	A mnemonic for the function code from the message header.
0/51	The VPI/VCI of the VC (on the control interface) on which the message was sent or received.

debug vsi param-groups

Use the **debug vsi param-groups** EXEC command to display the first 128 bytes of each VSI message sent and received by the TSC (in hexadecimal form). The **no** form of this command disables debugging output.

Note **param-groups** stands for parameter groups. A parameter group is a component of a VSI message.

[no] debug vsi param-groups [interface *interface* [slave *number*]]

Syntax Description

interface <i>interface</i>	Interface number.
slave <i>number</i>	Slave number (beginning with zero).

Usage Guidelines

This command is most commonly used with the **debug vsi packets** command to monitor incoming/outgoing VSI messages. If the interface parameter is specified, output is restricted to messages sent and received on the indicated VSI control interface. If the slave number is specified, output is further restricted to messages sent and received on the session with the indicated slave.

Note Slave numbers are the same as the session numbers discussed under the **show controllers vsi session** command.

Multiple uses of the form of the command, which specifies slave number, allows multiple slaves to be debugged at once. For example, the following commands restrict output to that for messages received on atm2/0 for sessions 0 and 1, (but for no other sessions).

```
Router# debug vsi param-groups interface atm2/0 slave 0
Router# debug vsi param-groups interface atm2/0 slave 1
```

Sample Display

The following is an example of the display you see when you enter **debug vsi param-groups**:

```
Router# debug vsi param-groups

Outgoing VSI msg of 12 bytes (not including encap):
01.02.00.80 00.00.95.c2 00.00.00.00
Incoming VSI msg of 72 bytes (not including encap):
01.02.00.81 00.00.95.c2 00.0f.00.3c 00.10.00.08
00.01.00.00 00.00.00.00 01.00.00.08 00.00.00.09
00.00.00.09 01.10.00.20 01.01.01.00 0c.08.80.00
00.01.0f.a0 00.13.00.15 00.0c.01.00 00.00.00.00
42.50.58.2d 56.53.49.31
Outgoing VSI msg of 12 bytes (not including encap):
01.02.00.80 00.00.95.c3 00.00.00.00
Incoming VSI msg of 72 bytes (not including encap):
01.02.00.81 00.00.95.c3 00.0f.00.3c 00.10.00.08
00.01.00.00 00.00.00.00 01.00.00.08 00.00.00.09
00.00.00.09 01.10.00.20 01.01.01.00 0c.08.80.00
00.01.0f.a0 00.13.00.15 00.0c.01.00 00.00.00.00
42.50.58.2d 56.53.49.31
```

Table 20 lists the significant fields shown in this display.

Table 20 Debug VSI Param-Groups Command Field Description

Field	Description
Outgoing	The message was sent by the VSI master.
Incoming	The message was received by the VSI master.
bytes	The number of bytes in the message, starting at the VSI header, and excluding the link layer encapsulation.
01.02...	Up to the first 128 bytes of the message, in hexadecimal form.

System Error Messages

This section lists and describes Cisco IOS Tag Switching system error messages. The system software sends these error messages to the console (and, optionally, to a logging server on another system) during operation. Not all system error messages indicate problems with your system. Some are purely informational, while others may help diagnose problems with communications lines, internal hardware, or the system software.

How to Read System Error Messages

System error messages begin with a percent sign (%) and are structured as follows:

```
%FACILITY-SUBFACILITY-SEVERITY-MNEMONIC: Message-text
```

FACILITY is a code consisting of two or more uppercase letters that indicate the facility to which the message refers. A facility can be a hardware device, a protocol, or a module of the system software. Table 21 lists the system facilities codes.

SEVERITY is a single-digit code from 0 to 7 that reflects the severity of the condition. The lower the number, the more serious the situation. Table 22 lists the severity levels.

MNEMONIC is a code that uniquely identifies the error message.

Message-text is a text string describing the condition. This portion of the message sometimes contains detailed information about the event, including terminal port numbers, network addresses, or addresses that correspond to locations in the system memory address space. Because the information in these variable fields changes from message to message, it is represented here by short strings enclosed in square brackets ([]). A decimal number, for example, is represented as [dec]. Table 23 lists the representations of variable fields and the type of information in them.

The following is a sample system error message:

```
%LINK-2-BADVCALL: Ints. TDR=[dec]
```

Table 21 Facility Codes

Code	Facility
AAA	TACACS+ Authentication, Authorization, and Accounting security
AIP	ATM Interface Processor
ALIGN	Memory optimization in Reduced Instruction-Set Computer (RISC) processor
AMDP2	Presidio Ethernet & Laguna Fast Ethernet
APPN	Advanced Peer-to-Peer Networking
ARAP	Apple Remote Access Protocol
ASPP	Asynchronous Security Protocol
AT	AppleTalk
ATM	Asynchronous Transfer Mode
BAP	PPP Bandwidth Allocation Protocol (BAP)
BGP	Border Gateway Protocol
BRI	Integrated Services Digital Network (ISDN) Basic Rate Interface

Table 21 Facility Codes (Continued)

Code	Facility
BRIMUX	AS5200 BRIMUX board
BSC	Binary Synchronous Communications mode
BSTUN	Block serial tunneling
C1600	Cisco 1600 platform
C3600	Cisco 3600 platform
C5RSP	Cisco Catalyst 5000 platform
CBUS	ciscoBus controller
CDM	Cable Data Modem subsystem
CI	75xx platform chassis interface
CIP facility	Channel Interface Processor
CIRRUS_PM	Slow speed async/sync port module
CLEAR	Clear facility
CLNS	OSI Connectionless Network Service
CLS	Cisco Link Services
CLSDR	Cisco Link Services Driver
COMP	Point-to-point compression
CONTROLLER	Controller
CPAD	Compression service adapter
CPM	Combo Port Module device driver
CSC2	CSC2/CSC3 CPU cards
CT3	Channelized T3 port adapter
DBUS	Data bus
DIALER	Dial-on-demand routing
DLC	Data-link control
DLSw	Data-link switching
DMA	Direct memory access
DNET	DECnet
DRP	Director Response Protocol
DSPU	Downstream physical unit
DSX1	Channelized E1 (Europe) and T1(US) telephony standard
DUAL	Enhanced Interior Gateway Routing Protocol
DVMRP	Distance Vector Multicast Routing Protocol
EGP	Exterior Gateway Protocol
ENT_API	Entity MIB API
ENV	Environmental monitor card
ETHERNET	Ethernet for the C1000 series
FDDI	Fiber Distributed Data Interface
FLASH	Flash nonvolatile memory

Table 21 Facility Codes (Continued)

Code	Facility
FR	Frame Relay
FTC_TRUNK	Cisco 3801 platform
GRIP	Xerox Network Systems (XNS) Routing Protocol
HD	HD64570 serial controller
HOOD	LAN controller 100VG-AnyLAN interface
HP100VG	100VG-AnyLAN PA driver
HUB	Cisco Ethernet hub
IBM2692	IBM Token Ring chip set
IFS	IOS File System
IGRP	Interior Gateway Routing Protocol
ILACC	ILACC driver
INTERFACE_API	Binary API for the interface descriptor block
IP	Internet Protocol
IPC	Interprocess Communication
IPFAST	IP fast switching
IPRT	Internet Protocol routing
IPX	Internetwork Packet Exchange Protocol
IP-SNMP	Simple Network Management Protocol specific to IP
ISDN	Integrated Services Digital Network
LANCE	Local Area Network Controller Ethernet
LANE	LAN Emulation
LANMGR	IBM LAN Network Manager
LAPB	X.25 Link Access Procedure, Balanced
LAT	DEC Local Area Transport
LEX	LAN extension
LINEPROTO	Line Protocol
LINK	Data link
LLC2	Logical Link Control type 2
LLIST	Linked list facility
LNMC	LAN network manager
LPD	Line printer daemon
MAILBOX	ChipCom mailbox support
MBRI	Multi-BRI port module
MCI	Multipoint Communications Interface
MK5	MK5025 serial controller
MPA68360	VIP Multi-channel Port Adapter
MROUTE	Multicast route
MUESLIX	Mx serial application-specific integrated circuit (ASIC)

Table 21 Facility Codes (Continued)

Code	Facility
NIC100	NIC100 driver
NIM	Network interface module
OSPF	Open Shortest Path First
PA	Port adapter
PAD	X.25 packet assembler/disassembler
PARSER	Parser
PIM	Protocol-independent multicast
PPP	Point-to-Point Protocol
QA	Queue and accumulator
QLLC	Qualified Logical Link Control
QUICC	MC68360 Quad Integrated Communications Controller
RADIUS	Remote Access Dial-In User Service (RADIUS) facility
RADIX	Radix facility
RCMD	Remote commands
RIP	IP Routing Information Protocol
RSP	Route Switch Processor
RSRB	Remote source-route bridging
S4T68360	Four port synchronous serial adapter based on the 68360 processor
SCHED	Scheduler
SDLC	Synchronous Data Link Control
SDLLC	SDLC/Logical Link Control type 2 (LLC2) translation
SEC	IP security
SERVICE_MODULE	Service Module
SLIP	Serial Line Internet Protocol
SMRP	Simple Multicast Routing Protocol
SNAPSHOT	Snapshot dial-on-demand routing
SNMP	Simple Network Management Protocol
SNMP_MGR	SNMP Proxy
SSE	Silicon switching engine
STANDBY	Hot Standby Router Protocol (HSRP)
STUN	Serial tunneling
SUBSYS	Software subsystems
SWITCH	Switch interface
SYS	Operating system
SYSMGT	System management
TAC	Terminal Access Controller Protocol Access Control System
TAGCON	Tag Switching Control

Table 21 Facility Codes (Continued)

Code	Facility
TCATM	Tag Controlled ATM
TDP	Tag Distribution Protocol
TE_LPDB	Traffic engineering loop prevention database
TFIB	Tag Forwarding Information Base
TIB	Tag Information Base
TRAFFENG	Traffic engineering
TSP_TUNNEL	Tag-Switched Path tunnel
TBRIDGE	Transparent bridging
TCP	Transmission Control Protocol
TMQ	Inbound terminal port queuing
TN	Telnet
TN3270	TN3270 protocol
TR	Token Ring
TUN	Tunnel
UCODE	Microcode
UNIX	UNIX
UTIL	Utility
VINES	Banyan VINES
VIP	Versatile Interface Processor
VPN	Virtual Private Dialup Network
VSI	Virtual Switch Interface
X25	X.25
XTAGATM	Extended Tag ATM interface

Table 22 Error Message Severity Levels

Level	Description
0 – emergency	System unusable
1 – alert	Immediate action needed
2 – critical	Critical condition
3 – error	Error condition
4 – warning	Warning condition
5 – notification	Normal, but significant, condition
6 – informational	Informational message only
7 – debugging	Appears during debugging only

Error message severity levels correspond to the keywords assigned by the **logging** global configuration commands that define where and at what level these messages appear. The default is to log messages to the console at the debugging level (7). For more information, see the system configuration chapter and descriptions of the **logging** command in the appropriate Cisco IOS configuration guide and command reference publications.

Table 23 Representation of Variable Fields in Error Messages

Representation	Type of Information
[atalk_address]	AppleTalk address
[atalk_net]	AppleTalk network, either 600 or 600-601
[char]	Single character
[chars]	Character string
[dec]	Decimal number
[enet]	Ethernet address (for example, 0000.FEED.00C0)
[hex]	Hexadecimal number
[inet]	Internet address (for example, 12.128.2.16)
[int]	Integer number
[node]	Address or node name
[sci_notation]	Scientific notation
[t-line]	Terminal line number in octal (or decimal if the decimal-TTY service is enabled)
[v-name]	VINES name; or number (hex or decimal)

Error Message Traceback Reports

Some messages describe internal errors and contain traceback information. This information is very important and should be included when you report a problem to your technical support representative.

The following sample message includes traceback information:

```
-Process= "Exec", level= 0, pid= 17  
-Traceback= 1A82 1AB4 6378 A072 1054 1860
```

Tag Switch Controller Error Messages

This section lists the Tag Switch Controller error messages.

Error Message %VSI_M-3-INCOMPATVER:None of the VSI versions ([int]-[int]) for session [dec] on [chars] can be used

Explanation The VSI master on the Tag Switch Controller, and the VSI slaves on the controlled switch, must all agree to use the same version of the VSI protocol. This error means that the slave for the indicated session on the indicated control interface does not support a VSI version which is also supported by the VSI master and all the other slaves. The session cannot be established while this is the case.

Recommended Action This error indicates that the Tag Switch Controller software version is not compatible with software on the controlled switch. Determine the software versions of the TSC and the switch software/firmware, and upgrade the software/firmware which is out of date.

Error Message

%VSI_M-2-XCONNFAIL:Cross-connect [chars]/[int]/[int] [chars] [chars]/[int]/[int] failed unexpectedly

Explanation One or more cross-connects previously established by the Tag Switch Controller (through VSI) have failed unexpectedly. The Tag Virtual Circuit of which this cross-connect was a part will no longer carry traffic end-to-end. The failure may be due to an unexpected condition on the switch which the TSC controls, or it may be due to an incorrect switch configuration performed while the switch was not in communication with the TSC, but before the TSC detected this lack of communication.

Recommended Action This error indicates one of more TVCs will no longer carry traffic end-to-end. Obtain the required information and contact your technical service representative.

Error Message

%XTAGATM-3-CONSISTENCY: [chars]

Explanation An action attempted by the XTagATM interface implementation encountered an unexpected condition

Recommended Action Copy the message exactly as it appears and report it to your technical service representative.

Error Message

%XTAGATM-3-CONTROLVC: [chars]; could not create the control VC; reason [chars]

Explanation The tag switching control VC could not be created on the specified extended tag atm (XTagATM) interface. This prevents a TDP session from being established on this interface.

Recommended Action This message indicates that the XTagATM driver was unable to create the tag switching control VC. This failure may be the result of a configuration problem or of an unexpected condition within the XTagATM driver. To check for a configuration problem, make sure that the VPI and VCI values for the tag control VC lie within the range supported on the controlled switch interface that is associated with the XTagATM interface (through the **extended-port** interface configuration command). If the configuration is correct, the problem may be due to an unexpected condition within the XTagATM driver. In this case, copy the message exactly as it appears, and report it to your technical service representative.

Error Message

%XTAGATM-3-CTLVCVPI:[chars]; VPI for control VC must be 0 or within configured range on switch

Explanation The VPI value for the tag control VC, set through the **tag-switching atm vpi** interface configuration command, must either be 0 or else must lie within the range of VPI values that is configured for Tag Switching on the controlled ATM switch. If the VPI value for the tag switching control VC does not satisfy these constraints, then the tag control VC is not created, and the TDP session is not established on the XTagATM interface.

Recommended Action Either reconfigure the VPI value for the tag switching control VC using the **tag-switching atm vpi...** interface configuration command on the Tag Switch Controller, or else reconfigure the range of VPI values available for Tag Switching on the controlled ATM switch. Make sure that the VPI value specified for the tag control VC is either 0 or else lies within the configured range of VPI values available for Tag Switching on the controlled ATM switch

Error Message

%XTAGATM-3-DUPEXTPORT:[chars]; mapped to the same switch interface as [chars]

Explanation Two XTagATM interfaces have been associated with a single interface on the controlled ATM switch through the **extended-port** interface configuration command. This causes one of the XTagATM interfaces to remain down indefinitely.

Recommended Action Use the **extended-port...** interface configuration command on one of the XTagATM interfaces to remove the duplicate binding.

Error Message

%XTAGATM-3-SWITCHVPI: [chars]; switch VPI range ([dec]-[dec]) outside usable range ([dec]-[dec])

Explanation The VPI range that has been configured for Tag Switching on the TSC-controlled ATM switch does not overlap with the range of VPI values that the TSC can support. This prevents a TDP session from being established on this interface.

Recommended Action Reconfigure the VPI range on the controlled switch so that it overlaps the usable range of VPI values.

What to Do Next

This document describes information specific only to the Tag Switch Controller. However, you can refer to the following publications for additional hardware and software configuration not specific to Tag Switch Controller.

- *Release Notes for Cisco IOS Release 11.1 CT and Feature Modules*
- *Cisco BPX 8600 Series Installation*
- *Cisco BPX 8600 Series Reference*