

## Chapter 7

# Switching

|                                                                     |      |
|---------------------------------------------------------------------|------|
| Introduction .....                                                  | 7-4  |
| Switch Ports .....                                                  | 7-5  |
| Enabling and Disabling Switch Ports .....                           | 7-5  |
| Speed and Duplex Mode .....                                         | 7-8  |
| Auto MDI/MDI-X .....                                                | 7-10 |
| Link Aggregation .....                                              | 7-12 |
| Link Aggregation Control Protocol (LACP) .....                      | 7-13 |
| Broadcast Storm Protection .....                                    | 7-14 |
| Port Mirroring .....                                                | 7-15 |
| Port Security .....                                                 | 7-16 |
| Limiting Rapid MAC Movement .....                                   | 7-17 |
| Support for Jumbo Frames .....                                      | 7-18 |
| Virtual Local Area Networks (VLANs) .....                           | 7-18 |
| Dynamic VLAN Assignment .....                                       | 7-19 |
| 802.1x Guest VLAN .....                                             | 7-21 |
| VLAN Tagging .....                                                  | 7-21 |
| VLAN Membership using VLAN Tags .....                               | 7-23 |
| VLAN Membership of Untagged Packets .....                           | 7-24 |
| Configuring VLANs .....                                             | 7-25 |
| The Default VLAN .....                                              | 7-28 |
| Static and Dynamic VLANs .....                                      | 7-29 |
| Private VLANs .....                                                 | 7-30 |
| Nested VLANs .....                                                  | 7-32 |
| The Layer 2 Switching Process .....                                 | 7-34 |
| The Ingress Rules .....                                             | 7-35 |
| The Learning Process .....                                          | 7-36 |
| The Forwarding Process .....                                        | 7-36 |
| The Egress Rules .....                                              | 7-37 |
| Layer 2 Filtering .....                                             | 7-38 |
| Layer 2 QoS Actions in Hardware Filters .....                       | 7-38 |
| Classifier-Based Packet Filters .....                               | 7-39 |
| Access Control Lists (ACLs) .....                                   | 7-40 |
| Classifier-Based Filters with Accelerated IPv6 Traffic .....        | 7-41 |
| Configuring Accelerator Hardware Filters for AT-8948 Switches ..... | 7-42 |
| Triggers .....                                                      | 7-42 |
| Configuration Examples .....                                        | 7-43 |
| Port-Based VLAN with Untagged Ports .....                           | 7-44 |
| VLAN with Tagged Ports .....                                        | 7-45 |
| Subnet-Based VLAN .....                                             | 7-47 |
| Command Reference .....                                             | 7-50 |
| activate switch port .....                                          | 7-50 |
| add lacp port .....                                                 | 7-51 |

|                                           |       |
|-------------------------------------------|-------|
| add switch accelerator hwfilter .....     | 7-52  |
| add switch filter action .....            | 7-54  |
| add switch hwfilter .....                 | 7-56  |
| add switch trunk .....                    | 7-57  |
| add vlan port .....                       | 7-58  |
| add vlan protocol .....                   | 7-62  |
| add vlan subnet .....                     | 7-64  |
| create switch trunk .....                 | 7-65  |
| create vlan .....                         | 7-68  |
| delete lacp port .....                    | 7-70  |
| delete switch accelerator hwfilter .....  | 7-70  |
| delete switch filter .....                | 7-71  |
| delete switch hwfilter .....              | 7-71  |
| delete switch trunk .....                 | 7-72  |
| delete vlan port .....                    | 7-73  |
| delete vlan protocol .....                | 7-75  |
| delete vlan subnet .....                  | 7-76  |
| destroy switch trunk .....                | 7-77  |
| destroy vlan .....                        | 7-77  |
| disable lacp .....                        | 7-78  |
| disable lacp debug .....                  | 7-78  |
| disable switch accelerator .....          | 7-78  |
| disable switch accelerator function ..... | 7-79  |
| disable switch accelerator debug .....    | 7-79  |
| disable switch ageingtimer .....          | 7-80  |
| disable switch debug .....                | 7-80  |
| disable switch hash .....                 | 7-81  |
| disable switch mclimiting .....           | 7-81  |
| disable switch learning .....             | 7-82  |
| disable switch mirror .....               | 7-82  |
| disable switch port .....                 | 7-83  |
| disable switch port vlan .....            | 7-84  |
| disable vlan debug .....                  | 7-85  |
| enable lacp .....                         | 7-85  |
| enable lacp debug .....                   | 7-85  |
| enable switch accelerator .....           | 7-86  |
| enable switch accelerator function .....  | 7-86  |
| enable switch accelerator debug .....     | 7-87  |
| enable switch ageingtimer .....           | 7-88  |
| enable switch bist .....                  | 7-88  |
| enable switch debug .....                 | 7-90  |
| enable switch hash .....                  | 7-91  |
| enable switch learning .....              | 7-92  |
| enable switch mclimiting .....            | 7-92  |
| enable switch mirror .....                | 7-92  |
| enable switch port .....                  | 7-94  |
| enable switch port vlan .....             | 7-95  |
| enable vlan debug .....                   | 7-96  |
| purge lacp .....                          | 7-96  |
| reset lacp port counter .....             | 7-97  |
| reset switch .....                        | 7-97  |
| reset switch accelerator counter .....    | 7-97  |
| reset switch port .....                   | 7-98  |
| set lacp port .....                       | 7-99  |
| set lacp .....                            | 7-100 |
| set switch ageingtimer .....              | 7-101 |
| set switch cputxpriorityoverride .....    | 7-101 |
| set switch cputxqueueoverride .....       | 7-102 |
| set switch dlflimit .....                 | 7-102 |

|                                        |       |
|----------------------------------------|-------|
| set switch enhancedmode .....          | 7-103 |
| set switch hwlearnndelay .....         | 7-104 |
| set switch hwrouteupdate .....         | 7-104 |
| set switch jumbo .....                 | 7-105 |
| set switch mirror .....                | 7-105 |
| set switch nestedtpid .....            | 7-106 |
| set switch port .....                  | 7-106 |
| set switch thrashlimit .....           | 7-112 |
| set switch trunk .....                 | 7-113 |
| set vlan port .....                    | 7-115 |
| show lacp .....                        | 7-116 |
| show lacp port .....                   | 7-117 |
| show lacp port counter .....           | 7-119 |
| show lacp trunk .....                  | 7-120 |
| show switch .....                      | 7-120 |
| show switch accelerator .....          | 7-125 |
| show switch accelerator counter .....  | 7-126 |
| show switch accelerator debug .....    | 7-129 |
| show switch accelerator hwfilter ..... | 7-130 |
| show switch counter .....              | 7-131 |
| show switch debug .....                | 7-133 |
| show switch fdb .....                  | 7-134 |
| show switch filter .....               | 7-135 |
| show switch hwfilter .....             | 7-137 |
| show switch port .....                 | 7-138 |
| show switch port counter .....         | 7-142 |
| show switch port intrusion .....       | 7-144 |
| show switch trunk .....                | 7-145 |
| show vlan .....                        | 7-147 |
| show vlan debug .....                  | 7-152 |
| show vlan port .....                   | 7-153 |

## Introduction

---

This chapter gives an overview of switching at Layer 1 (physical layer), Layer 2 (data link layer), and Layer 3 (network layer). Support for switching is also covered, along with how to configure and operate switching functions.

The switch, also referred to as a *MAC (media access control) bridge*, a *data link relay*, or a *Layer 3 switch*, can connect multiple Local Area Network (LAN) segments together to form an extended LAN. Stations connected to different LANs can be configured to communicate with one another as if they were on the same LAN. It can also divide one physical LAN into multiple Virtual LANs (VLANs). Stations connected to each other on the same extended LAN can be grouped in separate VLANs, so that a station in one VLAN can communicate directly with other stations in the same VLAN, but must go through higher layer routing protocols to communicate with stations in other VLANs.

The switch operates at the data link layer, transparent to higher layer protocols, transferring frames between the data link layers of the networks to which it is attached. A bridge accesses each physical link according to the rules for that particular network. Access may not always be instant, so a bridge must be capable of storing and forwarding frames. Since the switch can store and forward frames, it can examine and discard or admit frames according to their VLAN tag fields. The switch can also examine the address fields of the frames and forward the frames based on knowledge of which network contains the station with an address matching the frame's destination address. In this way, the switch can act as an intelligent filtering device, redirecting or blocking the movement of frames between networks.

Because the switch may receive frames faster than it can forward them, it has Quality of Service (QoS) queues where frames wait until the switch transmits them based on priority.

The switch can be used to:

- Increase the physical extent and/or the maximum number of stations on a LAN.

LANs are limited in their physical extent by the signal distortion and propagation delay characteristics of the media. The switch overcomes this limitation by receiving a frame on one LAN and then retransmitting the frame on another LAN, using the normal access methods for each LAN. The physical characteristics of the LAN media also place a practical limit on the number of stations that can be connected to a single LAN segment. The switch overcomes this limitation by joining LAN segments together to form an extended LAN capable of supporting more stations than either of the individual LANs.

- Connect LANs which have a common data link layer protocol but different physical media, for example, Ethernet 10BASET, 100BASET, and 10BASEF.
- Increase the availability of LANs by allowing multiple redundant paths to be physically configured, and selected dynamically, using the Spanning Tree algorithm.
- Reduce the load on a LAN or increase the effective bandwidth of a LAN, by filtering traffic.
- Prioritise the transmission of data with high Quality of Service requirements.

By using Virtual LANs (VLANs), a single physical LAN can be separated into multiple Virtual LANs. VLANs can be used to:

- Further improve LAN performance, as broadcast traffic is limited to LAN segments serving members of the VLAN to which the sender belongs.
- Provide security because frames are forwarded to only those stations belonging to the sender's VLAN, and not to stations in other VLANs on the same physical LAN.
- Reduce the cost of moving or adding stations to function on security-based LANs, as this generally requires only a change in the VLAN configuration.

## Switch Ports

---

*Port* is a common switch term. Each port in a switch is associated with a physical interface on the switch. Each port is uniquely identified by a number. The switch supports a number of features at the physical layer that allow it to be connected in a variety of physical networks. This Layer 1 versatility includes:

- [Enabling and Disabling Switch Ports](#)
- [Speed and Duplex Mode](#)
- [Auto MDI/MDI-X](#)
- [Link Aggregation](#)
- [Broadcast Storm Protection](#)
- [Port Mirroring](#)
- [Port Security](#)
- [Limiting Rapid MAC Movement](#)

## Enabling and Disabling Switch Ports

By default, all switch ports are enabled. An enabled port is available to receive and transmit packets. Its operational status and administrative status in the Interfaces MIB is **up**.

**Enabling ports** To enable a switch port, use the commands:

```
enable switch port={port-list|all} [other options]
```

A switch port that has been disabled by the Port Security feature ("[Port Security](#)" on page 7-16) cannot be enabled using the **enable switch port** command. Instead, use the [set switch port command on page 7-106](#) and set **learn=0**.

**Resetting ports** Resetting Ethernet ports at the hardware level discards all frames queued for reception or transmission on the port, restarts autonegotiation of port speed and duplex mode, and resets port counters. To reset ports, use the command:

```
reset switch port={port-list|all}
```

To reset port counters without resetting the ports, use the command:

```
reset switch port={port-list|all} counter
```

**Disabling ports** A disabled port is not available to receive and transmit packets. It does not send or receive any frames and its administrative status in the Interfaces MIB is **down**.

You can disable switch ports at the hardware or software level. Disabling a port at the hardware level has the same effect as physically removing the cable, GBIC, or SFP. Disabling a port at the software level only takes the link down in software.

We recommend disabling ports at the hardware level. This ensures that the port at the other end of the link realises that the port is down. To do this, use the command:

```
disable switch port={port-list|all} link=disable
```

To disable the port only at the software level, use the command:

```
disable switch port={port-list|all}
```

The following table further describes the options.

| If you...                                                             | by using...                                                | then...                                                          | and Status is... | and Link State is... |
|-----------------------------------------------------------------------|------------------------------------------------------------|------------------------------------------------------------------|------------------|----------------------|
| disable a port at the hardware level                                  | disable switch<br>port= <i>port-number</i><br>link=disable | port hardware and software link functionality both turn off      | DISABLED         | Down                 |
| disable a port at the software level (take the link down in software) | disable switch<br>port= <i>port-number</i>                 | software link functionality turns off but port hardware stays on | DISABLED         | Up                   |
| bring the link up in software without re-enabling the port            | disable switch<br>port= <i>port-number</i><br>link=enable  | software link functionality stays off but port hardware turns on | DISABLED         | Up                   |
| enable a port                                                         | enable switch<br>port= <i>port-number</i>                  | port hardware and software link functionality both turn on       | ENABLED          | Up                   |

### Displaying information

To display information about switch ports, use the command:

```
show switch port [= {port-list|all}]
```

The following figures show the possible combinations of status and link state for a port.

Figure 7-1: Output of the **show switch port** command when the port is enabled

```
Switch Port Information
-----
Port ..... 13
Description ..... -
Status ..... ENABLED
Link State ..... Up
.
.
.
```

Figure 7-2: Output of the **show switch port** command when the port is disabled at the software level

```
Switch Port Information
-----
Port ..... 13
Description ..... -
Status ..... DISABLED
Link State ..... Up
.
.
.
```

Figure 7-3: Output of the **show switch port** command when the port is disabled at the hardware level

```
Switch Port Information
-----
Port ..... 13
Description ..... -
Status ..... DISABLED
Link State ..... Down
.
.
.
```

## Speed and Duplex Mode

**Duplex mode** Ports can operate in full duplex or half duplex mode depending on the type of port it is. When in full duplex mode, a port transmits and receives data simultaneously. When in half duplex mode, the port transmits or receives but not both at the same time.

You can set a port to use either of these options, or allow it to autonegotiate the duplex mode with the device at the other end of the link.

**Speed options** x900-48FE and AT-9900 switches support ports with the following speeds:

- non-SFP RJ-45 copper ports on AT-8948 and x900-48FE switches: 10 or 100 Mbps
- copper or fibre SFPs on AT-8948 and x900-48FE switches: 1000 Mbps
- non-SFP RJ-45 copper ports on AT-9900 series switches: 10, 100 or 1000 Mbps
- supported tri-speed copper SFPs on AT-9900 series switches: 10, 100 or 1000 Mbps
- fibre SFPs on AT-9900 series switches: 1000 Mbps, with the exception of the 100 Mb fibre SFP on AT-9924SP models, which is 100 Mbps

x900-24X switches support ports with the following speeds:

- non-SFP RJ-45 copper ports: 10, 100 or 1000 Mbps
- supported tri-speed copper SFPs: 10, 100 or 1000 Mbps
- fibre SFPs: 100 and 1000 Mbps
- XFP modules: 10 Gbps

For the latest list of approved SFP transceivers either contact your authorised distributor or reseller, or visit [www.alliedtelesis.com](http://www.alliedtelesis.com).

You can set a port to use one of these speed options, or allow it to autonegotiate the speed with the device at the other end of the link.

**Autonegotiation** Autonegotiation lets the port adjust its speed and duplex mode to accommodate the device connected to it. When the port connects to another autonegotiating device, they negotiate the highest possible speed and duplex mode for both of them.

By default, all ports autonegotiate. Setting the port to a fixed speed and duplex mode may be necessary when connecting to a device that cannot autonegotiate.

**Configuring speed and duplex** To set the speed and duplex mode, use the command:

```
set switch port={port-list|all} [speed={autonegotiate|
10Mauto|10Mhalf|10Mfull|10Mhauto|10Mfauto|100Mauto|
100Mhalf|100Mfull|100Mhauto|100Mfauto|1000Mhalf|
1000Mfull|1000Mfauto}] [other-parameters]
```

The **speed** parameter combines speed, duplex mode, and autonegotiation support in a single setting. Options are in the following categories.

| Category      | Description                                                                                                                           |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------|
| autonegotiate | The <b>autonegotiate</b> option, which is the default.<br>If you specify this option, the port negotiates both speed and duplex mode. |



| Category                                  | Description (cont)                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fixed modes                               | <p>These options do not contain “auto”, such as <b>100Mfull</b>.</p> <p>If you specify one of these options, the port operates at that speed and duplex setting instead of autonegotiating with its link partner. For example, <b>100Mfull</b> means that the port transmits data at 100 Mbps full duplex mode.</p>                                                                                                                                                    |
| autonegotiate fixed speed and duplex mode | <p>These options contain a speed and duplex mode and “auto”, such as <b>100Mfauto</b>.</p> <p>If you specify one of these options, the port enters into autonegotiation with its link partner, but advertises that speed and duplex mode as the only mode it supports. For example, <b>100Mfauto</b> means that the port advertises that it can only support 100Mbps full duplex mode and <b>100Mhauto</b> means that it only advertises 100Mbps half duplex mode.</p> |
| autonegotiate fixed speed                 | <p>The <b>10Mauto</b> and <b>100Mauto</b> options.</p> <p>If you specify one of these options, the port enters into autonegotiation with its link partner, and negotiates the duplex mode but advertises that speed as the only speed it supports. For example, <b>100Mauto</b> means that the port advertises both half and full duplex mode at the one specified speed.</p>                                                                                          |

Make sure that the configuration of the switch matches the configuration of the device at the other end of the link. In particular, avoid having one end autonegotiate if the other end is fixed. For example, if you set one end of a link to **autonegotiate** and other to **100Mfull**, the autonegotiating end cannot determine that the fixed end is full duplex capable. Therefore, the autonegotiating end selects 100Mbps half-duplex operation. Using **100Mfauto** at the “fixed” end of the link would allow the autonegotiating end to autonegotiate 100Mbps full-duplex mode. This gains the benefits of autonegotiation while forcing operation at the desired speed.

Also, if you override a port’s autonegotiation by setting it to a fixed speed and duplex mode, automatic MDI/MDI-X detection is also overridden. The port defaults to MDI-X.

To display current speed and duplex mode settings, use the [show switch port command on page 7-138](#).

To activate autonegotiation at any time on ports that are set to autonegotiate, use the [activate switch port command on page 7-50](#).

**Port types and speed** Options for different types of ports are shown in the following tables.

| Switch               | Port type           | Speed Parameter Options                                                                                                      |
|----------------------|---------------------|------------------------------------------------------------------------------------------------------------------------------|
| AT-8948<br>x900-48FE | non-SFP RJ-45       | autonegotiate                                                                                                                |
|                      | copper ports        | 10Mauto, 10Mhauto, 10Mhalf, 10Mfauto, 10Mfull<br>100Mauto, 100Mhauto, 100Mhalf, 100Mfauto, 100Mfull                          |
|                      | copper SFPs         | autonegotiate<br>1000Mhalf, 1000Mfull, 1000Mfauto                                                                            |
|                      | fibre SFPs          | autonegotiate<br>1000Mfull, 1000Mfauto                                                                                       |
| AT-9900              | non-SFP RJ-45       | autonegotiate                                                                                                                |
|                      | copper ports        | 10Mauto, 10Mhauto, 10Mhalf, 10Mfauto, 10Mfull<br>100Mauto, 100Mhauto, 100Mhalf, 100Mfauto, 100Mfull<br>1000Mfull, 1000Mfauto |
|                      | supported tri-speed | autonegotiate                                                                                                                |
|                      | copper SFPs         | 10Mauto, 10Mhauto, 10Mhalf, 10Mfauto, 10Mfull<br>100Mauto, 100Mhauto, 100Mhalf, 100Mfauto, 100Mfull<br>1000Mfull, 1000Mfauto |
|                      | 100Mb fibre SFPs    | 100Mhalf, 100Mfull (AT-9924SP only)                                                                                          |
|                      | 1000Mb fibre SFPs   | autonegotiate<br>1000Mfull, 1000Mfauto                                                                                       |

| Port type for x900-24X          | Speed Options                                                                                                                                            |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| non-SFP RJ-45 copper ports      | autonegotiate<br>10Mauto, 10Mhauto, 10Mhalf, 10Mfauto, 10Mfull<br>100Mauto, 100Mhauto, 100Mhalf, 100Mfauto, 100Mfull<br>1000Mfull, 1000Mfauto            |
| supported tri-speed copper SFPs | autonegotiate<br>10Mauto, 10Mhauto, 10Mhalf, 10Mfauto, 10Mfull<br>100Mauto, 100Mhauto, 100Mhalf, 100Mfauto, 100Mfull<br>1000Mhalf, 1000Mfull, 1000Mfauto |
| 100Mb fibre SFPs                | 100Mhalf, 100Mfull                                                                                                                                       |
| 1000Mb fibre SFPs               | autonegotiate<br>1000Mfull, 1000Mfauto                                                                                                                   |
| XFP modules                     | autonegotiate<br>10G                                                                                                                                     |

## Auto MDI/MDI-X

MDI and MDI-X are Medium Dependent Interface port configurations for copper based interfaces. An MDI interface at one end and an MDI-X (MDI crossover) interface at the other end of a straight-through cable ensures the correct correspondence between the transmitting and receiving interface cable pairs. If both ends have MDI interfaces, or both ends have MDI-X interfaces, a crossover cable is required. *Polarity* refers to whether a port operates as MDI or MDI-X.

The auto MDI/MDI-X feature enables a switch port to detect whether it needs to have MDI or MDI-X polarity for a link. This means that straight-through (or crossover) cables can be used to connect these ports, whatever the MDI or MDI-X polarity of the other end.

Fixed switch ports and copper SFP transceivers plugged into the SFP sockets perform auto MDI/MDI-X by default as part of port autonegotiation. On fixed switch ports you can disable this auto MDI/MDI-X feature so that the ports are configured with a fixed MDI or MDI-X polarity. To disable auto MDI/MDI-X, use the command:

```
disable switch port={port-list|all} automdi
```

When auto MDI/MDI-X is disabled, the switch port polarity is set to the default, MDI-X. To modify the fixed polarity, use the command:

```
set switch port={port-list|all} polarity={mdi|mdix}  
[other-parameters]
```

Copper SFP transceivers plugged into the SFP sockets are auto MDI/MDI-X, and cannot be set to a fixed polarity. Fibre SFP ports do not have MDI/MDI-X polarity.

## Link Aggregation

Link aggregation, also known as *port bundling* or *port trunking*, allows a number of ports to be configured together to make a single logical connection of higher bandwidth. This can be useful where a higher performance link is required and it makes links even more reliable. Link aggregation must be configured on both ends of a link or network loops may result. Ports in a trunk group need not be contiguous. The following table explains ports in trunking groups.

| Switch    | Max No. of Trunks | Max No. of Ports per Trunk |
|-----------|-------------------|----------------------------|
| AT-8948   | 7                 | 4                          |
| x900-48FE | 7                 | 8                          |
| AT-9900   | 12                | 4                          |
| AT-9924Ts | 12                | 4                          |
| x900-24X  | 12                | 8                          |

The switch supports static 802.3ad link aggregation, and is also compatible with third party devices that do not support static 802.3ad link aggregation. It is not possible for a trunk group to include both 10/100 Ethernet and Gigabit Ethernet ports. A mirror port cannot be a member of a trunk group.

### How to configure

To create aggregation groups on the switch, use the [create switch trunk command on page 7-65](#). To destroy them, use the [destroy switch trunk command on page 7-77](#). Groups can be destroyed only when no ports belong to them.

The switch uses hashing selection criteria to determine which aggregation group a packet is transmitted across. Hashing is based upon Layer 2 and 3 header information in the packet.

All ports in a aggregation group must have the same VLAN configuration—they must belong to the same VLANs and have the same tagging status. All ports in a group must be added to VLANs together, and can only be deleted from a VLAN as a group. Similarly, if the tagged or untagged status of the ports changes, the change must be made for all ports in the aggregation group at the same time.

To display the VLANs to which the ports in a aggregation group belong, use the [show vlan command on page 7-147](#).

Members of a aggregation group can be specified when it is created, and ports are added with the [add switch trunk command on page 7-57](#). To remove them from a group, use the [delete switch trunk command on page 7-72](#).

Ports in a aggregation group are set to autonegotiate at the trunk speed at full duplex. When a port is added to an existing group, the speed setting for the group overrides the speed setting previously configured for the port. When a port is removed from a aggregation group, the port returns to its previously configured speed and duplex mode settings.

The speed of the aggregation group can be specified when it is created, or set later with the [set switch thrashlimit command on page 7-112](#).

To display information about trunks on the switch, use the [show switch trunk command on page 7-145](#).

To add an uplink aggregation group to a private VLAN or delete a port from a group that is the uplink ports of a private VLAN, see [“Private VLANs and port trunking” on page 7-32](#).

## Link Aggregation Control Protocol (LACP)

The implementation of the Link Aggregation Control Protocol (LACP) follows the IEEE Standard 802.3-2002, *CSMA/CD access method and physical layer specifications*.

LACP operates where systems are connected over multiple communications links. Once LACP has been initially configured and enabled, it automatically creates trunk groups and assigns appropriate links to their membership. LACP continues to monitor these groups and dynamically adds or removes links to them as network changes occur.

LACP achieves this by determining the following:

- which ports are under LACP control
- whether each port is in *LACP active* or *LACP passive* mode
- which system has the highest LACP priority
- the LACP priority of ports
- whether the periodic timeout is fast or slow

**Aggregation criteria** For individual links to be formed into an aggregated group they must meet the following criteria:

- originate on the same device
- terminate on the same device
- be members of the same VLANs
- have the same data rate
- share the same admin port key (assigned by using the **add lacp port adminkey** command)
- be operating in full duplex mode

The hardware must also be capable and have the capacity to handle the number of links to be aggregated.

**Aggregated group identification** In order to identify particular aggregated groups, each group is assigned a link aggregation identifier called a *lag ID*. The lag ID comprises the following components for both the local system (called the Actor) followed by their equivalent components for the remote system (called the Partner):

- *system priority* - set by the [set lacp command on page 7-100](#).
- *system identifier* - the MAC address of the system
- *port key* - An identifier - created by the LACP software
- *port priority* - set by the [add lacp port command on page 7-51](#).
- *port number* - determined by the device connection

The lag ID can be displayed for each aggregated link by entering the [show lacp trunk command on page 7-120](#).

## Broadcast Storm Protection

The storm control feature allows you to set limits on the reception rate of the following:

- broadcast and multicast packets on a per port basis, i.e one limit per port
- destination lookup packets for all ports on the switch, i.e one limit for the switch

The switch hardware counts separately the number of broadcast, multicast, and destination lookup failure packets in bytes received per second, and discards packets once the byte limit is reached.

The minimum rate is 100Kbytes per second, increasing in steps of 100Kbytes per second to the maximum rate of 95300Kbytes per second.

### Broadcast and multicast rate limiting

You can set one port limit for broadcast and multicast packet limiting. Therefore, the maximum reception rate limit is the same for both broadcast and multicast packets on a port, and independent limits for broadcast and multicast packets cannot be set.

You can enable the following on a per port basis:

- broadcast rate limiting
- broadcast and multicast rate limiting

If multicast rate limiting is enabled, then it is active for all ports that:

- already have broadcast limiting set and enabled
- subsequently have broadcast limiting enabled after multicast limiting is enabled

Multicast rate limiting is either on for all ports that have broadcast limiting enabled, or off for all ports.

To set broadcast storm control and the reception rate limit for a port, use the command:

```
set switch port=port-list bclimit={none|limit}
[other-parameters]
```

where *limit* is a decimal number between 100 and 95300, in Kbytes per second

To enable and disable the multicast storm control, which works with broadcast limiting, use the command:

```
enable switch mclimiting
disable switch mclimiting
```

To display whether broadcast and multicast storm control is enabled or disabled for a port, or all ports, use the command:

```
show switch port=port-list
```

### Destination lookup failure packets

Destination lookup failure packets have a Layer 2 destination address that the switch has not learned and are in effect multicast packets. The switch does not know where to forward the packets so the packets are broadcast to all ports on the switch. You can limit the rate at which destination lookup failure packets are received.

To set destination lookup failure rate limiting on the switch, use the command:

```
set switch dlflimit={none|limit}
```

where *limit* is a decimal number between 100 and 95300, in Kbytes per second

To display whether destination lookup failure storm control is enabled, use the command:

```
show switch
```

## Port Mirroring

Port mirroring allows traffic received and transmitted on a switch port to be sent to another switch port—the mirror port—usually to capture the data with a protocol analyser. Either the traffic received or the traffic transmitted by the port, or both, can be mirrored. Incoming and outgoing traffic need not be mirrored to the same port.

Before the mirror port can be set, it must be removed from all VLANs. The port cannot be part of a trunk group. To set the mirror port (and remove it from the default VLAN) use the command:

```
set switch mirror={none|port}
```

If another port was previously set as the mirror port, this command returns the previous mirror port to the default VLAN as an untagged port. Return this port to any VLAN to which it should belong by using the [add vlan port command](#) on page 7-58.



**Caution** Mirroring four or more ports may significantly reduce switch performance.

Port mirroring is disabled by default and no mirror port is set. Mirroring can be enabled after the switch mirror port has been set to a valid port. If mirroring has been enabled but the switch mirror port is set to **none**, then mirroring is disabled.

A maximum of 8 egress ports and/or 8 ingress ports can be mirrored.

To enable and disable mirroring, use the commands:

```
enable switch mirror
```

```
disable switch mirror
```

To set mirroring, use the command:

```
set switch port={port-list|all} [mirror={both|none|rx|tx}]  
[other-options]
```

The commands **show switch** and **show switch port** display the switch and switch port mirroring settings.

## Port Security

The port security feature allows control over the stations connected to each switch port, by MAC address. If enabled on a port, the switch learns MAC addresses up to a user-defined limit from 1 to 320, then locks out other MAC addresses. One of the following options can be specified for the action taken when an unknown MAC address is detected on a locked port:

- discard the packet and take no further action
- discard the packet and notify management with an SNMP trap
- discard the packet, notify management with an SNMP trap, and disable the port

To enable port security on a port, set the limit of learned MAC addresses to a value greater than zero, and specify the action to take for unknown MAC addresses on a locked port. To disable port security on a port, set the limit of learned MAC addresses to zero or **none**. To enable or disable security on a port, use the command:

```
set switch port={port-list|all} learn={none|0|1..320}
[intrusionaction={discard|trap|disable}]
[relearn={off|on}] [other-options]
```

If **intrusionaction** is set to **trap** or **disable**, a list of MAC addresses for devices that are active on a port, but which are not allowed or learned for the port, can be displayed by using the command:

```
show switch port={port-list|all} intrusion
```

When **relearn** is **on**, Dynamic Port Security is used. Dynamic Port Security enables dynamic MAC address learning. If a MAC address is unused for a period of time, it is aged from the database of currently accepted MAC addresses. This enables new MAC addresses to be learned.

The **relearn** parameter defaults to **off**. In this case static MAC address learning is used. This means that MAC addresses once learned, cannot be unlearned.

To manually lock a switch port before it reaches the learning limit, use the command:

```
activate switch port={port-list|all} lock
```

Addresses can be manually added to a port locked list up to a total of 320 MAC addresses, and the learning limit can be extended to accommodate them by using the command:

```
add switch filter action={forward|discard} destaddress=macadd
port=port [entry=entry] [learn] [vlan={vlan-name|1..4094}]
```

Learned addresses on locked ports can be saved as part of the switch configuration and become part of the configuration after a power cycle. Use the command:

```
create config=filename
```

If the configuration is not saved when there is a locked list for a port, the learning process begins again after the switch restarts. Addresses learned through dynamic MAC learning are not added to the configuration.

A switch port that has been disabled by the Port Security feature cannot be enabled using the **enable switch port** command. To re-enable such a port, set its learn limit to zero.



## Limiting Rapid MAC Movement

MAC address thrashing occurs when MAC addresses move rapidly between one or more ports or trunks, for example, due to a network loop.

Limiting rapid MAC movement is supported on all port types. It is also supported on trunked ports. This is configured using the [create switch trunk command on page 7-65](#) and the [set switch thrashlimit command on page 7-112](#).

**Disabling a port** There are different ways you can disable a port when thrashing is detected. These are called *thrash actions* and are:

- **learnDisable**  
Address learning is temporarily disabled on the port.
- **portDisable**  
The port is logically disabled. Traffic flow is prevented, but the link remains up. The device at the other end does not notice that the port has changed status, and the link LEDs at both ends stay on. This is equivalent to entering the **disable switch port** command.
- **linkDown**  
The port is physically disabled and the link is down. This is equivalent to entering the **disable switch port link=disabled** command.
- **vlanDisable**  
The port is disabled only for the VLAN on which thrashing has occurred. It can still receive and transmit traffic for any other VLANs of which it is a member.

When a MAC address is thrashing between two ports, only one of those ports is disabled. When multiple ports are involved, enough ports are disabled to prevent the storm.

To set a thrash action for a port, use the command:

```
set switch port={port-list|all} [thrashaction={learndisable|linkdown|none|portdisable|vlandisable}]
```

To view the thrash action that is set for a port, use the command:

```
show switch port={port-list|all}
```

To set a thrash action for a trunk, use one of the commands:

```
create switch trunk=trunk [port=port-list]
[thrashaction={learndisable|linkdown|none|portdisable|vlandisable}]
```

```
set switch thrashlimit=trunk [thrashaction={learndisable|linkdown|none|portdisable|vlandisable}]
```

To view the thrash action that is set for a trunk, use the command:

```
show switch trunk={trunk}
```

To view details about disabled ports for VLANs, use one of the commands:

```
show vlan[={vlan-name|1..4094|all}]
show vlan[=all]
```

**Re-enabling a port** When a port is disabled, either completely or for a specific VLAN, it remains disabled until it is manually re-enabled in any of the following ways:

- with SNMP
- as the result of a reboot
- by specifying a thrash timeout value along with the thrash action
- via the CLI

If the **vlandisable** thrash action has been applied, then to re-enable one or more ports from VLANs to which they belong, use the command:

```
enable switch port={port-list|all} vlan[={vlan-name|1..4094|all}]
```

If either the **portdisable** or **linkdown** thrash action has been applied, then to re-enable one or more ports, use the command:

```
enable switch port={port-list|all}
```

If the **learndisable** thrash action has been applied, the port is automatically re-enabled when the defined timeout expires. You cannot manually re-enable the port.

## Support for Jumbo Frames

You can enable jumbo frame support on the switch to improve throughput and decrease CPU utilization. By increasing frame size, more data is put in each packet that the switch has to process.

When jumbo frames support is enabled, the maximum received packet size is:

- 9710 bytes for ports that work at speeds of either 10Mbps or 100Mbps
- 10240 bytes for ports that work at speeds of 1000Mbps

Jumbo frame support is enabled or disabled on the entire switch, not on a per port basis. To enable them, use the [set switch jumbo command on page 7-105](#); to see whether they are enabled, use the [show switch command on page 7-120](#). They are disabled by default.

## Virtual Local Area Networks (VLANs)

A Virtual LAN (VLAN) is a logical, software-defined subnetwork. It allows similar devices on the network to be grouped together into one broadcast domain, regardless of their physical position in the network. Several VLANs can be configured so that workstations, servers, and other network equipment connected to the switch can be grouped according to similar data and security requirements. The switch has 4094 user-configurable VLANs.

The ability to decouple logical broadcast domains from the physical wiring topology offers several advantages, including the ability to:

- move devices and people with minimal reconfiguration or none
- change a device's broadcast domain and access to resources without physically moving the device, by software reconfiguration or by moving its cable from one switch port to another
- isolate parts of the network from other parts, by placing them in different VLANs

- share servers and other network resources without losing data isolation or security
- direct broadcast traffic to only those devices which need to receive it, to reduce traffic across the network
- connect 802.1Q-compatible switches together through one port on each switch

Devices that are members of the same VLAN exchange data only with each other through the switch's switching capabilities. To exchange data between devices in separate VLANs, the switch uses routing capabilities. The switch passes VLAN status information, indicating whether a VLAN is up or down, to the Internet Protocol (IP) module. IP uses this information to determine route availability.

There are a maximum of 4094 VLANs on the switch that range from the VLAN identifier (VID) of 1 to 4094.

When the switch is first powered up, a "default" VLAN is created and all ports are added to it. This VLAN has a VID of 1. It cannot be deleted, and ports can only be removed from it if they also belong to at least one other VLAN. If all devices on the physical LAN are to belong to the same logical LAN, meaning the same broadcast domain, then the default settings are acceptable. No additional VLAN configuration is required. For more information about the default VLAN, see ["The Default VLAN" on page 7-28](#).

## Dynamic VLAN Assignment

Dynamic VLAN assignment allows a supplicant to be placed into a specific VLAN based on information returned from the RADIUS server during authentication. This limits the network access of a supplicant to a specific VLAN that is tied to their authentication, and prevents supplicants from connecting to VLANs for which they are not authorised. A port's VLAN assignment is determined by the first supplicant to be authenticated on the port.

VLAN assignment is enabled or disabled using the **vlanassignment** parameter of a number of port authentication commands. For more information, see [Chapter 32, Port Authentication](#).

The Configured and Actual fields of the **show vlan** command show which ports are configured for the VLAN and which have been dynamically assigned to the VLAN.

### RADIUS attributes

The RADIUS server provides information to the authenticator using RADIUS tunnel attributes, as defined in RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*. The tunnel attributes that must be configured for VLAN assignment are:

- **Tunnel-Type**  
The protocol to be used for the tunnel specified by Tunnel-Private-Group-Id. VLAN (13) is the only supported value.
- **Tunnel-Medium-Type**  
The transport medium to be used for the tunnel specified by Tunnel-Private-Group-Id. 802 (6) is the only supported value.
- **Tunnel-Private-Group-ID**  
The ID of the tunnel the authenticated user should use. This must be the name or ID number of a VLAN on the switch.

These tunnel attributes are included in the Access-Accept message from the RADIUS server to the Authenticator.

### Single-host mode

In single host mode, VLAN assignment is as follows:

- If authentication fails, the supplicant is denied access to the port. The port is placed in its configured access VLAN, that is, the VLAN it was set up for in the **add vlan** command.
- If the RADIUS server supplies valid VLAN information, the port is placed in the specified VLAN after configuration.
- If the RADIUS server supplies invalid VLAN information, the port is returned to the Unauthorised state, and placed in its configured access VLAN.
- If the RADIUS server supplies no VLAN information, the port is placed in its configured access VLAN after successful authentication.
- If port authentication is disabled on the port, the port is returned to its configured access VLAN.
- When the port is in the Force Authorized, Force Unauthorized or the Unauthorized state, it is placed in its configured access VLAN.

While the port is in a RADIUS server assigned VLAN, changes to the port's configured access VLAN do not take effect until the port leaves the assigned VLAN. This can occur if:

- the last authentication session on the port expires
- the link goes down
- port authentication is disabled on the port
- port authentication is disabled on the system

### Multi-suppliant mode

VLAN assignment can be run in multi-suppliant mode. In multi-suppliant mode, the behaviour is dictated by which supplicant is authenticated first.

If the multi-suppliant mode is enabled on a port authentication port, the behaviour of the first authenticated supplicant is the same as that of a supplicant in single-suppliant mode. For all further supplicants, the **securevlan** parameter specifies the action that is taken when authenticating supplicants after the first supplicant has authenticated.

| If securevlan is... Then... |                                                                                                                                                                          |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| on                          | Only those supplicants with a VLAN that is the same as that of the first authenticated supplicant are authenticated. This is the default, and is the more secure action. |
| off                         | All further authenticated supplicants are placed in the same VLAN as the first authenticated supplicant. This action is less secure.                                     |

## 802.1x Guest VLAN

802.1x ports can be configured with a limited access guest VLAN, which is used when no 802.1x host is currently attached to the port. This limited access VLAN is defined using the **guestvlan** parameter.

As soon as a single 802.1x packet is received on the port, it is removed from the guest VLAN, and put into its configured access VLAN in the Unauthenticated state. This effectively disables the guest VLAN on the port until the port's link goes down.

A guest VLAN can only be configured for a port that is running in single-suplicant mode.

## VLAN Tagging

An Ethernet packet can contain a *VLAN tag* with fields that specify VLAN membership and user priority. The VLAN tag is described in IEEE Standard 802.3ac, and is four octets that can be inserted between the Source Address and the Type/Length fields in the Ethernet packet ([Figure 7-4](#)). To accommodate the tag, Standard 802.3ac also increased the maximum allowable length for an Ethernet frame to 1522 octets (the minimum size is 64 octets). IEEE Standard 802.1Q specifies how the data in the VLAN tag is used to switch frames. VLAN-aware devices are able to add the VLAN tag to the packet header. VLAN-unaware devices cannot set or read the VLAN tag.

The meaning and use of the fields in the Ethernet frame is listed in [Table 7-1](#). The format of VLAN data in an Ethernet frame is shown in [Figure 7-4 on page 7-22](#). Twelve bits of the tag are the VLAN identifier (VID), which indicates to which VLAN the packet belongs. The VLAN identifier values that have specific meaning are listed in [Table 7-2 on page 7-22](#).

Table 7-1: Fields in the Ethernet frame for QoS and VLAN switching

| Field         | Length   | Meaning and use                                                                                                                                                                                                             |
|---------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TPID          | 2 octets | The Tag Protocol Identifier (TPID) is defined by IEEE Standard 802.1Q as 0x81-00.                                                                                                                                           |
| User Priority | 3 bits   | The user priority field is the priority tag for the frame, which can be used by the switch to determine the Quality of Service to apply to the frame. The three bit binary number represents eight priority levels, 0 to 7. |
| CFI           | 1 bit    | The Canonical Format Indicator (CFI flag) indicates whether all MAC address information that may be present in the MAC data carried by the frame is in canonical format.                                                    |
| VID           | 12 bits  | The VLAN identifier (VID) field uniquely identifies the VLAN to which the frame belongs.                                                                                                                                    |

Figure 7-4: Format of user priority and VLAN data in an Ethernet frame

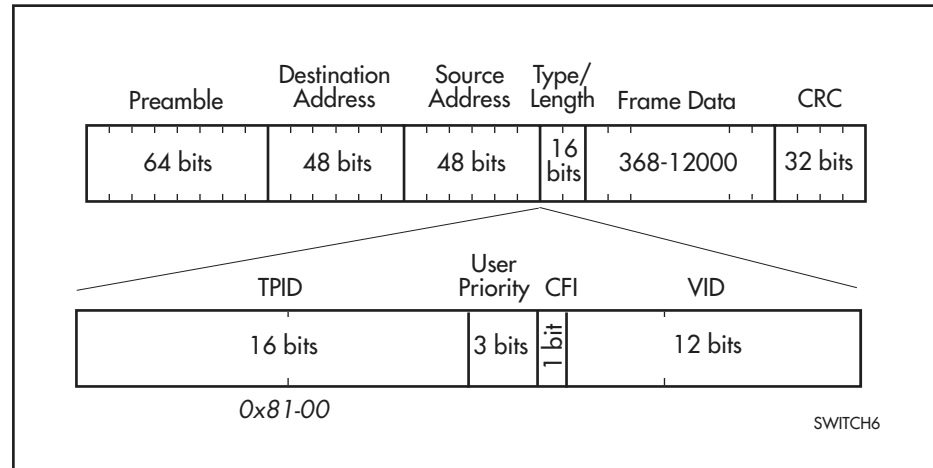


Table 7-2: Reserved VID values

| VID value<br>(hexadecimal) | Meaning and use of reserved VID values                                                                                                                                                                                                                                                                                               |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0                          | Null VLAN ID. Indicates that the tag header contains only user priority information; no VLAN identifier is present in the frame. This VID value must not be configured in any forwarding database entry, or used in any management operation. Frames that contain the null VLAN ID are also known as <i>priority-tagged frames</i> . |
| 1                          | Default VID used to classify frames on ingress through an untagged switch port.                                                                                                                                                                                                                                                      |
| FFF                        | Reserved for implementation use. This VID must not be configured in any forwarding database entry, used in any management operation, or transmitted in a tag header.                                                                                                                                                                 |

Ethernet packets which contain a VLAN tag are referred to as *tagged frames*, and switch ports that transmit tagged frames are referred to as *tagged ports*. Ethernet packets which do not contain the VLAN tag are referred to as *untagged frames*, and switch ports that transmit untagged frames are referred to as *untagged ports*. VLANs can consist of simple logical groupings of untagged ports, in which the ports receive and transmit untagged packets. Alternatively, VLANs can contain only tagged ports, or a mixture of tagged and untagged ports.

The switch is VLAN-aware. It can accept VLAN tagged frames, and supports the VLAN switching required by such tags. A network can contain a mixture of VLAN-aware devices, for example, other 802.1q-compatible switches, and VLAN-unaware devices, for example, workstations and legacy switches that do not support VLAN tagging. The switch can be configured to send VLAN tagged or untagged frames on each port, depending on whether the devices connected to the port are VLAN-aware. By assigning a port to two different VLANs, to one as an untagged port and to another as a tagged port, it is possible for the port to transmit both VLAN-tagged and untagged frames. A port must belong to a VLAN at all times unless the port has been set as the mirror port for the switch.

For information about double tagging, see [“Nested VLANs” on page 7-32](#).

## VLAN Membership using VLAN Tags

Ports can belong to many VLANs as tagged ports. Therefore, when the VLAN tag is used to determine which VLAN a packet belongs to, it is easy to:

- share network resources, such as servers and printers, across several VLANs
- configure VLANs that span several switches

A network configured with VLAN tagging is shown in [Figure 7-5 on page 7-23](#). [Table 7-3 on page 7-23](#) shows the VLAN membership. The server on port 2 on Switch A belongs to both the *admin* and *marketing* VLANs. The two switches are connected through port 5 on Switch A and port 24 on Switch B, which belong to both the *marketing* VLAN and the *training* VLAN, so devices on both VLANs can use this link.

Figure 7-5: Example using tagged ports

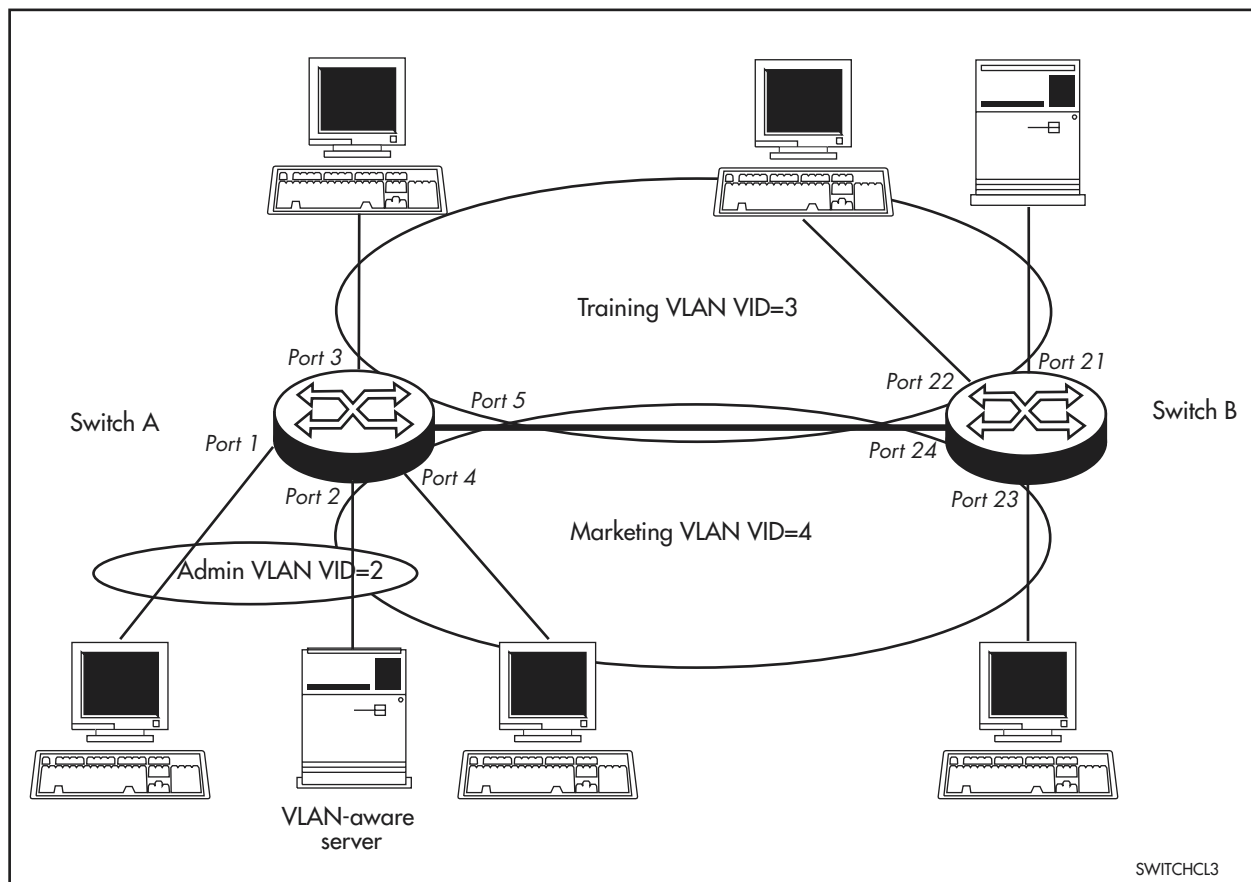


Table 7-3: VLAN membership of example of a network using tagged ports

| VLAN      | Member ports                               |
|-----------|--------------------------------------------|
| Training  | 3, 5 on Switch A<br>21, 22, 24 on Switch B |
| Marketing | 2, 4, 5 on Switch A<br>23, 24 on Switch B  |
| Admin     | 1, 2 on Switch A                           |

## VLAN Membership of Untagged Packets

The switch determines the VLAN that incoming untagged packets belong to based on classifications described in the following table. Because VLANs are multi-typed, a VLAN can have all three classifications.

| Classification | Precedence | Description                                                                                        |
|----------------|------------|----------------------------------------------------------------------------------------------------|
| IP subnet      | Highest    | For IP packets, the subnet where the packet came from.                                             |
| Protocol       | Medium     | The protocol of the packet.                                                                        |
| Port           | Lowest     | The port where the packet arrives. See <a href="#">“Port-Based Classifications” on page 7-24</a> . |

If the switch receives a packet that matches more than one of the associations for different VLANs, it is processed in order of precedence. For example, consider a switch with two VLANs. VLAN2 is configured so that packets from the subnet 192.168.1.0 are classified as belonging to it. VLAN3 is configured so that packets with protocol type IP are classified as belonging to it. If a port belongs both VLAN2 and VLAN3 and receives an IP packet from subnet 192.168.1.0, the packet belongs to VLAN2. If the port receives a packet from any other subnet, the packet belongs to VLAN3.

### Port-Based Classifications

The simplest VLAN classification is by port. This is the default VLAN classification. A port-based VLAN has a list of ports belonging to it, and all untagged traffic arriving at those ports belongs to that VLAN. Port-based VLANs are limited because each port can belong to only one port-based VLAN as an untagged port. Limitations include:

- It is difficult to share network resources, such as servers and printers, across several VLANs. The routing functions in the switch must be configured to interconnect using untagged ports only.
- A VLAN that spans several switches requires a port on each switch for the interconnection of the various parts of the VLAN. If there are several VLANs in the switch that span more than one switch, then many ports are occupied with connecting the VLANs, and so are unavailable for other devices.

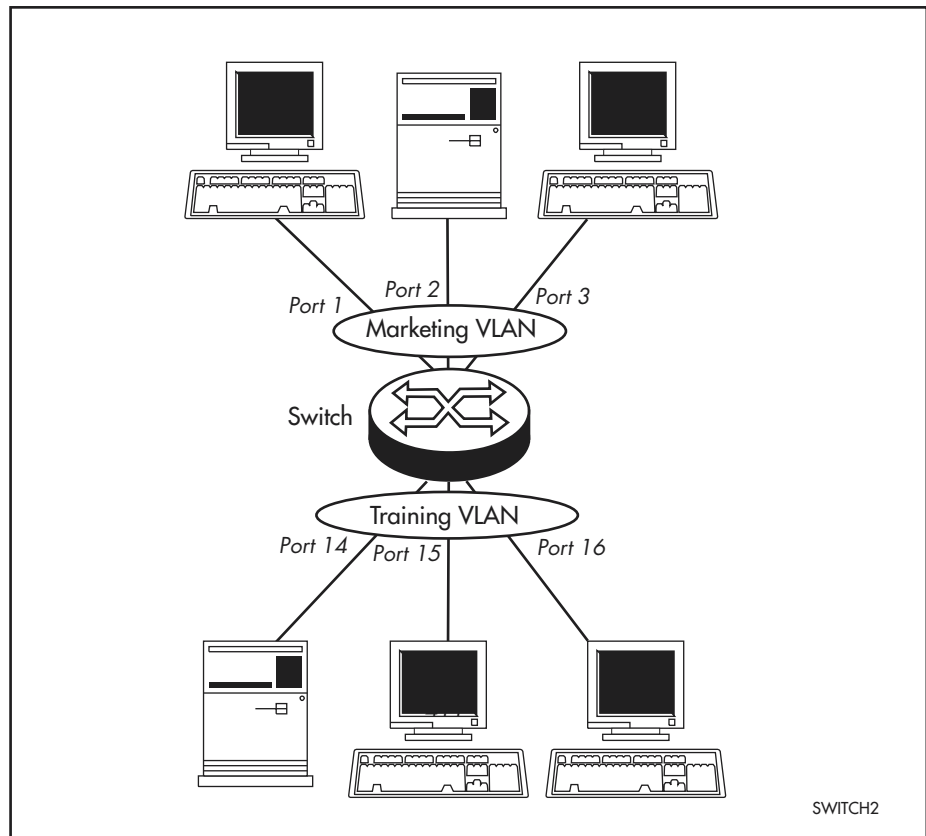
If the network includes VLANs that do not need to share network resources or span several switches, these VLANs can usefully be port-based. Otherwise, VLAN membership should be determined by tagging (see [“VLAN Tagging” on page 7-21](#)) or by one of the fields of untagged packets (see [“VLAN Membership of Untagged Packets” on page 7-24](#)).

Each untagged port must be associated with the port-based association of a VLAN. By default, all ports belong to the port-based association of the default VLAN, until they are manually removed from it.

Two port-based VLANs with untagged ports belonging to them are shown in [Figure 7-6 on page 7-25](#). Ports 1-3 belong to the *marketing* VLAN, and ports 14-16 belong to the *training* VLAN. The switch acts as two separate bridges: one that forwards traffic between the ports belonging to the *marketing* VLAN, and a second one that forwards traffic between the ports belonging to the *training* VLAN. Devices in the *marketing* VLAN can communicate with devices in the *training* VLAN by using the switch's routing functions.



Figure 7-6: VLANs with untagged ports



## Configuring VLANs

This section describes how to configure a VLAN. The procedure in summary is:

1. Create the VLAN and specify its classification, either IP subnet, protocol, or port.
2. Add tagged ports to the VLAN, if required.
3. Create associations to associate subnets and protocols with the VLAN if untagged ports are required. These associations determine to which VLAN that incoming untagged packets belong.
4. Add untagged ports to the associations.

### Creating a VLAN and specifying the VLAN classification

Up to 4094 VLANs can be created, with VIDs ranging from 2 to 4094. VLANs do not have to be numbered consecutively. Creating a VLAN involves (in a single **create vlan** command) giving the VLAN a name, specifying a VID for the VLAN, and specifying its classification. A particular subnet or protocol can be associated with the VLAN at the same time, or later with the **add vlan** command.

When the switch first powers up and the default VLAN is created, a port classification rule is automatically associated with it, and all ports are added to this association. This does not restrict the VLANs that can be created.

To create an IP subnet-based VLAN, use the command:

```
create vlan=vlan-name vid=2..4094 [subnet=ipadd] [mask=ipadd]
```

If an IP subnet is specified with the **subnet** and **mask** parameters, packets from the subnet are classified as belonging to this VLAN. Alternatively, a subnet (or more subnets) can be added later (see [“VLAN associations” on page 7-26](#)).

To create a protocol-based VLAN, use the command:

```
create vlan=vlan-name vid=2..4094 [protocol=protocol-type]
```

If a protocol is specified with the **protocol** parameter, packets of that protocol type are classified as belonging to this VLAN. Alternatively, a protocol type (or more protocol types) can be added later ([“VLAN associations” on page 7-26](#)).

To create a port based VLAN, use the command:

```
create vlan=vlan-name vid=2..4094
```

## Adding tagged ports to VLANs

Tagged ports use the VID in the VLAN tag of incoming packets to determine to which VLAN the packet belongs. Because the switch is VLAN-aware, it sets the VID field of the VLAN tag to the appropriate VLAN when it transmits packets.

To add tagged ports to a VLAN, use the command:

```
add vlan={vlan-name|1..4094} port={port-list|all}
frame=tagged
```

A port cannot transmit tagged and untagged traffic for the same VLAN.

If a port is associated with a VLAN's port-based association as an untagged port, it can be changed to a tagged port by using the command:

```
set vlan={vlan-name|1..4094} port={port-list|all}
frame=tagged
```

## VLAN associations IP subnet-based

After a VLAN has been created, IP subnets and/or protocols can be associated with it by using one of the following commands:

```
add vlan={vlan-name|1..4094} subnet=ipadd [mask=ipadd]
add vlan={vlan-name|1..4094} protocol=protocol-type
```

Packets received from the specified subnet, or of the specified protocol, are classified as belonging to the VLAN.

Subnet ranges cannot overlap. So if a subnet is associated with a VLAN, a subset or superset of that subnet cannot be associated with that VLAN again, or with another VLAN.

### Protocol-based

After a VLAN has been created, protocols can be associated with it by using the command:

```
add vlan={vlan-name|1..4094} protocol=protocol-type
```

Packets of the specified protocol type are classified as belonging to the VLAN.

When any VLAN is created, an empty port association is automatically created. This allows untagged ports to be added to the VLAN without associating them with a subnet or protocol (see [“Adding untagged ports to VLANs” on page 7-27](#)).

Ports in a single VLAN can have different VLAN associations. The different associations on different ports represent the rules by which packets arriving on those ports are associated with the VLAN. However, separate rules for separate sets of ports does not mean that the VLAN has been divided into separate broadcast domains. To split ports into separate broadcast domains, put ports in separate VLANs.

## Adding untagged ports to VLANs

Untagged traffic is classified as belonging to a VLAN when it matches one of the VLAN's associations, *and* arrives on a port that has been added to that VLAN association as an untagged port.

Traffic from each combination of association type and port can be classified as belonging to only one VLAN. For example, if IP traffic arriving on port2 is associated with VLAN2, it cannot also be associated with VLAN3.

To add untagged ports to the port-based association that was automatically created when the VLAN was created, use the command:

```
add vlan={vlan-name|1..4094} port={port-list|all}
    frame=untagged
```

If the port is not associated with another VLAN, all untagged traffic received on the port is classified as belonging to the specific VLAN. A port must belong to the port association of zero or to one VLAN.

If a port already belongs to a VLAN as a tagged port, it can be added to the port based association as an untagged port instead, using the command:

```
set vlan={vlan-name|1..4094} port={port-list|all}
    frame=untagged
```

To add untagged ports to a subnet-based association or associations, use the command:

```
add vlan={vlan-name|1..4094} port={port-list|all}
    subnet={ipadd|all}
```

To add untagged ports to a protocol-based association or associations, use the command:

```
add vlan={vlan-name|1..4094} port={port-list|all}
    protocol={protocol-type|index-list|all}
```

## Deleting untagged ports

To delete untagged ports from a VLAN association, use one of the commands:

```
delete vlan={vlan-name|1..4094} port={port-list|all}

delete vlan={vlan-name|1..4094} port={port-list|all}
    subnet={ipadd|all}

delete vlan={vlan-name|1..4094} port={port-list|all}
    protocol={protocol-type|index-list|all}

delete vlan={vlan-name|1..4094} port={port-list|all}
    [group=groupnumber|uplinknumber]
```

When a port belongs to the specified associations for the specified VLAN, the port is deleted from the VLAN. When the deleted port does not belong to a static VLAN as a tagged port, and the port is disassociated from the port association of the VLAN, then the port is implicitly added to the default VLAN as an untagged port and associated with the port association of the default VLAN. It is not possible to delete a port that belongs only to the default VLAN.

Associations can be deleted only when no ports belong to them. To delete associations from a VLAN, use one of the commands:

```
delete vlan={vlan-name|1..4094} subnet={ipadd|all}

delete vlan={vlan-name|1..4094} protocol={protocol-type|
    index-list|all}
```

To delete tagged ports from a VLAN, use the command:

```
delete vlan={vlan-name|1..4094} port={port-list|all}
```

If the deletion means that the port is no longer a member of a VLAN, it is added as an untagged port to the port association of the default VLAN.

Ports tagged for some VLANs and left in the default VLAN as untagged ports transmit broadcast traffic for the default VLAN. If this is not wanted, the unnecessary traffic in the switch can be reduced by deleting those ports from the default VLAN.

A VLAN cannot be destroyed when ports still belong to it or if other modules except GARP are attached to it. To destroy VLANs, use the command:

```
destroy vlan={vlan-name|2..4094|all}
```

The default VLAN, which has a VID of 1, cannot be destroyed. Therefore, if **all** is specified, all static VLANs except the default VLAN are destroyed.

### Link aggregation

All ports in a link aggregation (port trunk) must have the same VLAN configuration. Ports must belong to the same VLANs and have the same tagging status because they are handled as a group. For more information, see [“Link Aggregation” on page 7-12](#)).

### Displaying information about VLANs

To display VLANs configured on the switch, use the commands:

```
show vlan[={vlan-name|1..4094}]
```

```
show vlan[=all]
```

```
show vlan[={vlan-name|1..4094|all}] port[={port-list|all}]
```

Information which may be useful for trouble-shooting a network is displayed with VLAN debugging mode. VLAN debugging mode is disabled by default, and can be enabled for a specified time, disabled, and displayed by using the commands:

```
enable vlan={vlan-name|1..4094|all} debug={pkt|all}
[output=console] [timeout={1..4000000000|none}]
```

```
disable vlan={vlan-name|1..4094|all} debug={pkt|all}
```

```
show vlan debug
```

To display packet reception and transmission counters for a VLAN, use the command:

```
show interface=vlan n counter
```

See the [show interface command on page 10-51 of Chapter 10, Interfaces](#) for more information about this command.

## The Default VLAN

The default VLAN is created automatically when the switch is first powered up, and all the ports on the switch are added. In this initial unconfigured state, the switch broadcasts all packets it receives to the default VLAN. The default VLAN has a VID of 1 and its interface name is vlan1. The default VLAN cannot be destroyed.

When a port is removed from other VLANs, it is added to the default VLAN's port association as an untagged port unless the port becomes a mirror port. A port can be removed from the default VLAN only when it belongs to another VLAN.

The initial configuration of the default VLAN is unclassified. The VLAN can be given a classification type of IP subnet or protocol by adding an association to it. Use the commands:

```
add vlan=1 subnet=ipadd [mask=ipadd]  
add vlan=1 protocol=protocol-type
```

If the default VLAN is the first VLAN on the switch to be classified, the classification it receives limits the classification of other VLANs created. If another VLAN is created before the default VLAN is given a classification type, this sets the default VLAN's classification.

Ports in the default VLAN can be added as untagged ports to another VLAN as described in [“Adding untagged ports to VLANs” on page 7-27](#). If a port in the default VLAN is added to another VLAN's port association, the port is automatically removed from the default VLAN. If required, the port can be added as an untagged port to the default VLAN again but only to a non-port association.

For example, port 1 is in the default VLAN. To configure the port so that traffic addressed to subnet 192.168.1.1 goes to VLAN2 and all other traffic goes to the default VLAN:

1. Create VLAN2 and give it a subnet association for the subnet 192.168.1.1 by using the command:

```
create vlan=vlan2 vid=2 subnet=192.168.1.1  
mask=255.255.255.0
```

2. Add port 1 to this association by using the command:

```
add vlan=vlan2 port=1 subnet=192.168.1.1
```

Port 1 is now untagged for both the port association of the default VLAN, and the subnet=192.168.1.1 association of VLAN2.

## Static and Dynamic VLANs

All VLANs created by the user on the command line are static VLANs. The default VLAN is also a static VLAN. A port must belong to at least one static VLAN.

Dynamic VLANs are created by GVRP, a GARP application whose purpose is to propagate VLAN information between VLAN aware switches (see [Chapter 9, Generic Attribute Registration Protocol \(GARP\)](#)). Dynamic VLANs are named `gvrpxxx`, where `xxx` is the VLAN's VLAN identifier. Dynamic VLANs are created only when GVRP is enabled on the switch. GVRP is disabled by default.

The user can destroy all static VLANs except for the default VLAN. Dynamic VLANs cannot be directly destroyed by the user, but may be destroyed according to the operations of GVRP by using the [reset `garp` command on page 9-14 of Chapter 9, Generic Attribute Registration Protocol \(GARP\)](#), or by disabling the GVRP instance.

The user can add, delete, or modify ports belonging to any static VLAN. However, the user cannot add, delete, or modify ports to a dynamic VLAN. Dynamic VLANs created by GVRP include tagged ports.

## Private VLANs

A private VLAN contains switch ports that cannot communicate with each other but can access another network. These ports are called *private ports*. Each private VLAN contains one or more private ports, and a single uplink port or uplink trunk group.

A typical application is at a hotel where each room has a port that can access the Internet. In this situation it is undesirable to allow communication between rooms. Another application is to simplify IP address assignment. Ports can be isolated from each other while belonging to the same subnet.

The switch forwards all traffic that is received on a private port out that VLAN's uplink port, regardless of VLAN ID or MAC destination address. Packets received on an uplink port are forwarded in the normal way (i.e. as for non-private VLANs) for all types of packets.

The following table summarising action taken on packets received from private and uplink ports.

| For this packet... | Action if received on private port... | Action if received on uplink port...                                                              |
|--------------------|---------------------------------------|---------------------------------------------------------------------------------------------------|
| Unicast            | Forwarded out uplink port             | Destination port determined on basis of MAC address (as for non-private VLANs)                    |
| Multicast          | Forwarded out uplink port             | Forwarded out all private ports in the VLAN that are members of the multicast group for that VLAN |
| Broadcast          | Forwarded out uplink port             | Forwarded out all private ports in the VLAN                                                       |

Note that all traffic between private ports is blocked at all layers, not only Layer 2 traffic.

It is not possible to use protocols, such as Telnet or SNMP, to manage the switch via private ports. Use either non-private ports or uplink ports to manage the switch.

The switch cannot act as a DHCP server to hosts connected to private ports. However, a DHCP server running on a device connected to a private port's uplink port can act in this role.

### Membership rules

Each private VLAN:

- must contain one uplink port or uplink trunk group
- may contain multiple private ports
- can be configured to span switch instances
- cannot contain any non-private ports
- cannot be the default VLAN (vlan1)

Each private port:

- can be a member of multiple private VLANs, but all these VLANs must have the same uplink port or uplink trunk group
- cannot be a private port in some VLANs and a non-private port in other VLANs
- cannot be an uplink port in another VLAN

Each uplink port:

- can be a member of multiple private VLANs
- cannot be a member of both private and non-private VLANs

Each private or uplink port:

- may use any VLAN classification rule (port, subnet or protocol)
- may be tagged or untagged, but can only be an untagged member of one port-based VLAN
- may be trunked

## Configuring private VLANs

### To create a private VLAN and add ports to it

#### 1. Create the VLAN.

To create a VLAN and specify that it is private, use the command:

```
create vlan=vlan-name vid=2..4094 private
```

#### 2. Add the uplink port or trunk group.

To add the uplink ports to a private VLAN, use one of the commands:

```
add vlan={vlan-name|1..4094} port={port-list}
[frame={untagged|tagged}] uplink

add vlan={vlan-name|1..4094} port={port-list} subnet={ipadd|all} uplink

add vlan={vlan-name|1..4094} port={port-list}
protocol={protocol-type|index-list|all} uplink
```

where *portlist* is either a single port number for a single uplink port, or a list of port numbers for a trunk group. If you are adding a trunk group to the VLAN as an uplink, the ports must already be trunked together, and you must specify all the ports.

#### 3. Add the private ports.

Private ports can be added only after the uplink port or ports have been added to the private VLAN.

To add private ports to a private VLAN, use one of the commands:

```
add vlan={vlan-name|1..4094} port={port-list|all}
[frame={untagged|tagged}]

add vlan={vlan-name|1..4094} port={port-list|all}
subnet={ipadd|all}

add vlan={vlan-name|1..4094} port={port-list|all}
protocol={protocol-type|index-list|all}
```

### To delete ports from a private VLAN

To delete private ports from a private VLAN, use one of the commands:

```
delete vlan={vlan-name|1..4094} port={port-list|all}

delete vlan={vlan-name|1..4094} port={port-list|all}
subnet={ipadd|all}

delete vlan={vlan-name|1..4094} port={port-list|all}
protocol={protocol-type|index-list|all}
```

A private VLAN cannot contain any private ports when an uplink port is deleted from the VLAN because a private VLAN must always contain one uplink port.

To delete the uplink port from a private VLAN, first delete any private ports from the VLAN, then use one of the previous commands.

## Private VLANs and port trunking

All ports in a trunk group must have the same VLAN configuration.

Follow these rules for private VLANs and port trunking:

- The port list cannot include any uplink ports when creating a trunk group
- The ports cannot be a mix of private ports and non-private ports when creating a trunk group
- Multiple uplink ports can be added to a private VLAN as a trunk group

To add an uplink trunk group to a private VLAN:

1. Create a trunk group.
2. Add all of the trunk group's ports to the VLAN as uplink ports.

You cannot delete a port from a trunk group if the trunk group is the uplink ports of a private VLAN. To delete a port from such a trunk group:

1. Remove the private ports from the VLAN.
2. Remove the uplink trunk group from the VLAN.
3. Remove the port or ports from the trunk group.
4. Add the uplink trunk group to the VLAN.
5. Add the private ports to the VLAN.

For a full description of port trunking, see [“Link Aggregation” on page 7-12](#).

## Nested VLANs

Nested VLANs, also known as *VLAN double tagging*, are used to overlay a private Layer 2 network over a public Layer 2 network. This provides simple access and infrastructure for network service providers in Metropolitan Area Networks (MANs).

A nested VLAN implementation consists of:

- Core ports - connected to a service provider (public Layer-2 network)
- Customer ports - connected to a customer (private Layer-2 network)
- Customer-IDs (CIDs) - a second VLAN identifier (VID) assigned to each customer

### How nested VLANs work

When nested VLAN functionality is enabled, each customer is given a CID, consisting of a 12-bit identifier, within the service provider network.

The CID is attached to a packet while the packet is transmitted within the service provider, or public Layer-2, network. Within the service provider network the VID is ignored and bridging is based on the value of the CID. The ethertype of the CID is set by changing the Tag Protocol Identifier (TPID).

Any VLAN classification rule - Protocol, Subnet, or the default port-based - can be used to assign the CID. Note that subnet-based VLAN assignment does not work for non-IP packets, for example ARP packets, and we recommend that subnet-based VLAN classification not be used for nested VLANs.

Modification of the CID occurs on the Core port. On ingress the CID is removed and on egress the CID is inserted into the packet. Packets received and transmitted on Customer ports are unmodified.



Classifier-based packet filters, configured in hardware, have precedence over VLAN classification rules. If a classifier-based packet filter is attached to any ingress port, either Core or Customer, it overrides the bridging decision and the packet is forwarded according to the packet filter configured for that port.

Customer ports are automatically added as untagged ports to the nested VLAN. Core ports are automatically added as tagged ports.

Once the CID is removed from the packet it is forwarded “as is” out of the Customer port. The tagged status of the Customer port is ignored on egress.

### Rules for nested VLANs

When nested VLANs are created on the switch:

- a nested VLAN belongs to no more than one customer and can have multiple Customer ports
- a port must be either a Customer or Core port, but cannot be both
- a port cannot be both a “normal” port and a nested port

A Core port:

- accepts only tagged packets
- transmits only tagged packets
- can be in many nested VLANs

A Customer port:

- accepts both tagged and untagged packets
- transmits both tagged and untagged packets
- can be in only one nested VLAN

### Restrictions when using nested VLANs

Restrictions when nested VLANs are implemented are:

- Ethernet bridging is based on the CID instead of the packet VID. The packets VID does not change
- IGMP snooping is not available
- ARP packet trapping is restricted.
- routing and MPLS switching are not supported due to the non-standard packet header
- hardware filtering does not work above MAC address level

### Configuring nested VLANs

To create a nested VLAN and add ports to it

#### I. Create the VLAN.

To create a VLAN and specify that it is nested, use one of the commands:

```
create vlan=vlan-name vid=2..4094 nested
create vlan=vlan-name vid=2 4094 subnet=ipadd [mask=ipadd]
nested
create vlan=vlan-name vid=2..4094 protocol=protocoltype
nested
```

## 2. Add ports to the nested VLAN.

To add Customer ports to the nested VLAN, use the command:

```
add vlan={vlan-name|1..4094} port={port-list}
    nestedtype=customer
```

to add core ports to the nested vlan, use the command:

```
add vlan={vlan-name|1..4094} port={port-list} nestedtype=core
```

## 3. Set the Tag Protocol Identifier (TPID).

To set the Tag Protocol Identifier (TPID) for packets transmitted out of core ports, use the command:

```
set swi nestedtpid=tag-number
```

The **nestedtpid** parameter specifies the Ethernet type of the tagged packet. This is set to 0x8100 by default when a nested VLAN is created.

Note that this command specifies the TPID value that applies to all VLANs used for nested VLANs. The TPID value cannot be set for only one particular VLAN if more than one nested VLAN is created.

## To delete ports from a nested VLAN.

Use one of the commands:

```
delete vlan={vlan-name|1..4094} port={port-list|all}
```

```
delete vlan={vlan-name|1..4094} port={port-list|all}
    subnet={ipadd|all}
```

```
delete vlan={vlan-name|1..4094} port={port-list|all}
    protocol={protocol-type|index-list|all}
```

# The Layer 2 Switching Process

The Layer 2 switching process comprises these related but separate processes:

- [The Ingress Rules](#)
- [The Learning Process](#)
- [The Forwarding Process](#)
- [The Egress Rules](#)

Ingress rules admit or discard frames based on their VLAN tagging.

The Learning process learns the MAC addresses and VLAN membership of frames admitted on each port.

The Forwarding process determines which ports the frames are forwarded to, and the Quality of Service priority with which they are transmitted.

Finally, Egress rules determine for each frame whether VLAN tags are included in the Ethernet frames that are transmitted.

These processes assume that each station on the extended LAN has a unique data link layer address, and that all data link layer frames have a header which includes the source (sender's) MAC address and destination (recipient's) MAC address.

## The Ingress Rules

All frames, tagged and untagged, that a VLAN-aware switch receives must be associated with a VLAN. Each received frame is mapped to exactly one VLAN. If an incoming frame is tagged with a valid VLAN identifier (VID) then that VID is used. If an incoming frame is untagged or is priority tagged (a tagged frame with a VID of all zeros), then the switch uses internal VLAN association rules to determine the VLAN it belongs too. The default setting for the ingress rules is to Admit All Frames.

The possible association rules, in order of precedence, are:

- IP subnet/IPX network classification
- protocol classification
- port classification

The switch supports a subset of these VLAN association rules at any one time. The possible subsets used are:

- IP subnet, protocol and port classification
- protocol and port classification
- port classification

The default VLAN association is based upon the port on which the incoming untagged, or priority tagged, frame was received. It is possible for an incoming untagged, or priority tagged, frame to match more than one of the association rules. In such a case, the order of precedence is determined by the order of entries in the VLAN Tag table. Matching entries that appear first in the VLAN Tag table have the highest precedence.

The Acceptable Frame Types parameter controls the reception of VLAN-tagged and untagged frames on a port. Each port on the switch can be configured to admit:

- only untagged frames
- only VLAN-tagged frames
- both untagged and VLAN-tagged frames

A port that transmits only VLAN-tagged frames, regardless of which VLAN the port belongs to, automatically admits only VLAN-tagged frames. These ports should be used to connect VLANs between VLAN-aware switches, or to connect a VLAN-aware device to a VLAN-aware end station.

A port that transmits untagged frames for a VLAN must either be associated with one of the VLAN classification rules or is associated with the default Port VLAN classification rule. For example, an untagged port that belongs to an IP subnet-based VLAN must be associated with one IP subnet or one protocol belonging to the VLAN, or be associated with the Port VLAN classification rule of the VLAN. A port that transmits untagged frames can belong to only one port-based VLAN.

Incoming untagged frames may be classified into different VLANs. We recommend that if the user requires VLANs spanning multiple VLAN-aware switches, then outgoing frames must contain tag headers when they are destined for other VLAN-aware switches. This is particularly important when the port that frames are received on is a member of two or more VLANs. Adding a tag header to an outgoing frame is the only guaranteed method to ensure that other VLAN-aware switches can correctly identify the VLAN to which the frame should be associated.

## The Learning Process

The learning process uses an adaptive learning algorithm, sometimes called *backward learning*, to discover the location of each station on the extended LAN.

All frames admitted by the ingress rules on any port are passed on to the forwarding process when they are for destinations in the same VLAN. Frames destined for other VLANs are passed to a Layer 3 protocol, such as IP. For every frame admitted, the frame's source MAC address and VID are compared with entries in the forwarding database for the VLAN (also known as a *MAC Address table*) maintained by the switch. When the frame's source address is not in the forwarding database for the VLAN, the address is added and an ageing timer for that entry is started. When the frame's source address is already in the forwarding database, the ageing timer for that entry is restarted.

By default, switch learning is enabled. It can be disabled with the [disable switch learning command on page 7-82](#). Enable it again with the [enable switch learning command on page 7-92](#).

If the ageing timer for an entry in the forwarding database expires before another frame with the same source address is received, the entry is removed from the forwarding database. This prevents the forwarding database from being filled with information about stations that are inactive or have been disconnected from the network. It also ensures that entries for active stations are kept alive in the forwarding database.

By default, the ageing timer is enabled. It can be disabled with the [disable switch ageingtimer command on page 7-80](#). Enable it again with the [enable switch ageingtimer command on page 7-88](#).

If switch learning is disabled and the ageing timer has aged out all dynamically learned filter entries, only statically entered MAC source addresses decide the packets to forward or discard. When the switch finds no matching entries in the forwarding database during the forwarding process, all switch ports in the VLAN are flooded with the packet, except the port that received it.

The default of the ageing timer is 300 seconds (5 minutes) and can be modified by using the command:

```
set switch ageingtimer=10..630
```

To display general switch settings, including settings for switch learning and the switch ageing timer, use the [show switch command on page 7-120](#).

## The Forwarding Process

After a VID is assigned to a frame using the ingress rules, the switch forwards it to the destination MAC address specified in the frame. To do this the switch must learn which MAC addresses are available on each port for each VLAN. When the destination MAC address is not found, the switch floods the frame on all ports that are members of the VLAN except the port on which the frame was received. The member ports of a VLAN are stored in the outports mask field of the VLAN broadcast table (also known as the *Layer 2 broadcast table*).

The forwarding database (also known as the *MAC Address table*) determines the egress port on which the destination MAC address has been learned. MAC addresses are either learned dynamically during Layer 2 switching, or learned statically by user configuration, by the operations of a Layer 2 protocol, or by the Port Security feature.

The forwarding database is ordered according to MAC address and VLAN identifier. This means a MAC address can appear more than once in the forwarding database having been learned on the same port but for different VLANs. This could occur if the IP address of an end station is changed thereby moving the end station to a different IP subnet-based VLAN while still connected to the same switch port. When the forwarding database ageing process is enabled, old entries in the forwarding database are deleted after a user-configurable period.

The forwarding process provides storage for queued frames to be transmitted over a specific port. Each port has 8 Quality of Service transmission queues available. See [Chapter 27, Quality of Service \(QoS\)](#) for more information.

If the destination address is found, the switch discards the frame when the port is not in the STP forwarding or disabled state if the destination address is on the same port as the source address, or if there is a static filter entry for the destination address set to **discard** (see [“Layer 2 Filtering” on page 7-38](#)). Otherwise, the frame is forwarded on the indicated port.

Forwarding occurs only when the port on which the frame was received is in the Spanning Tree forwarding or disabled state. The destination address is then looked up in the forwarding database for the VLAN.

## The Egress Rules

After the forwarding process has determined from which ports and transmission queues to forward a frame, the egress rules for each port determine whether the outgoing frame is VLAN-tagged with its numerical VLAN identifier (VID).

A port must belong to a VLAN at all times unless the port has been set as the mirror port for the switch.

A port can transmit VLAN-tagged frames for any VLAN to which the port belongs. A port can transmit untagged frames for any VLAN for which the port is configured, e.g. IP subnet-based or protocol-based, unless prevented by the port-based VLAN egress rules. A port that belongs to a port-based VLAN can transmit untagged packets for only one VLAN. For more information about VLANs and VLAN tagging, see [“Virtual Local Area Networks \(VLANs\)” on page 7-18](#).

When a port is added to a port-based VLAN it can be configured to transmit either tagged or untagged frames using the command:

```
add vlan={vlan-name|1..4094} port={port-list|all}
    frame={tagged|untagged}
```

This setting can be changed for a port which is already part of a VLAN, using the command:

```
set vlan={vlan-name|1..4094} port={port-list|all}
    frame={tagged|untagged}
```

This command changes the status of ports in a port-based VLAN from tagged to untagged, or from untagged to tagged.

## Layer 2 Filtering

The switch has a forwarding database (also known as the *MAC Address table*) whose entries determine whether frames are forwarded or discarded over each port. Entries in the forwarding database are created dynamically by the learning process. A dynamic entry is automatically deleted from the forwarding database when its ageing timer expires.

The user can configure static switch filter entries using the command line interface. Static switch filter entries associate a MAC address with a VLAN and a port in the VLAN. When the switch receives a frame with a destination address and VLAN identifier that match those of a static filter entry, the frame can be either forwarded to the port specified in the static filter entry, or discarded.

The forwarding database supports queries by the forwarding process as to whether frames with given values of the destination MAC address field should be forwarded to a given port.

To add or delete a static switch filter entry, use the command:

```
add switch filter action={forward|discard} destaddress=macadd
port=port [entry=entry] [learn] [vlan={vlan-name|1..4094}]

delete switch filter port=port entry=entry-list
```

To display current static and learned switch filter entries, use the command:

```
show switch filter [port={port-list|all}]
[destaddress=macadd] [entry=entrylist] [vlan={vlan-name|
1..4094}]
```

For each VLAN, the destination MAC address of a frame to be forwarded is checked against the forwarding database. If there is no entry for the destination address and VLAN, the frame is transmitted on all ports in the VLAN that are in the forwarding or disabled state, except the port on which the frame was received. This process is referred to as *flooding*. If an entry is found in the forwarding database but the entry is not marked *forwarding* or the entry points to the same port the frame was received on, the frame is discarded. Otherwise, the frame is transmitted on the port specified by the forwarding database.

## Layer 2 QoS Actions in Hardware Filters

You can classify traffic and use a hardware filter to set its queue, 802.1p user priority or bandwidth class. This is a mechanism for elevating the probability of CPU reception for packets that you determine to be “important”.

In heavily congested networks, data streams can sometimes use up all the available bandwidth of the CPU receive process. This increases the probability that infrequently-sent packets are lost, for example, routing protocol packets (BGP, OSPF, PIM, DVMRP) or STP packets. By creating an appropriate classifier and hardware filter, such packets can be given preferential treatment, in terms of CPU reception.

To configure the hardware filter, use the command:

```
add switch hwfilter=filter-id classifier=rule-id
action=setl2qos [l2qosqueue=0..7] [priority=0..7]
[bandwidthclass=1..3]
```

The **action** parameter specifies what action the classifier entry takes with packets that match the classifier. If you specify **setl2qos**, then packets matching

the classifier have their bandwidth class (or drop precedence), queue, and 802.1p user priority modified to the values specified by the **bandwidthclass**, **l2qosqueue**, and **priority** parameters. The action **setl2qos** cannot be specified at the same time as **drop**.

The **l2qosqueue** parameter specifies to which queue to send matching packets. The default is 0.

The **priority** parameter specifies the 802.1p user priority with which to remark matching packets. The default is 0.

The **bandwidthclass** parameter specifies to which bandwidth class (drop precedence) to assign matching packets. The default is 1.

## Classifier-Based Packet Filters

The switch hardware can be configured through entries in the Packet Classifier to copy, drop, or forward packets that match specified criteria ([Chapter 26, Generic Packet Classifier](#)).

Every packet passing through the switch is matched against a series of classification tables by the Packet Classifier. You can classify packets according to many features, including:

- Packet type
- Layer 3 protocol
- Source/destination IP address
- Destination IPX address
- Layer 4 protocol (for example, TCP/UDP/Socket number)
- Layer 4 source/destination ports
- Source/Destination MAC address
- Source VLAN

To filter by destination MAC address, use the [add switch filter action command on page 7-54](#).

The user can configure classifier-based packet filters to take action upon the results of the classification tables. These actions are:

- Discard the packet
- Forward the packet
- Copy the packet to the CPU
- Copy the packet to the CPU with either discard or forward
- Set Layer 2 quality of service parameters

Classifier-based packet filters are numbered from 1 to 1024. The number of packet filters supported by the switch is determined by the amount of available space in the packet classification tables.

A classifier-based packet filter comprises a number of ordered classifier entries. The user can choose how classifiers are associated with packet filters. Classifiers can be grouped in one or more packet filters, or associated separately with a single packet filter.

Packet filters with low filter IDs have precedence over those with high ones. For example, a packet is matched against entries in the packet filter 1 before being matched against entries in packet filter 2. Likewise for the order of classifier entries within a packet filter. Packets are matched against the order of entries in the classifier table and then action is taken by the packet filter on the first match found.

If a later entry is subsumed by an earlier entry within the classifier table, both entries are valid, but the later entry is never used. For example, if packet filter 1 specifies that all IP packets are to be discarded irrespective of source port and packet filter 2 specifies all packets received on port 4 are to be forwarded, packet filter 2 will not function properly. Because packet filter 1 has precedence over packet filter 2, packets are first matched against the classifier entries in packet filter 1, action is taken against any packet match, then a match is made against classifier entries in packet filter 2 and action is taken.

**Important** The order in which packet filters are configured and the order in which classifier entries in each packet filter are configured are **critical** for the performance of classifier-based packet filtering.

The drop and forward actions are the core functions of the classifier-based packet filters implemented by the switch.

To add packet filters to the switch, use the command:

```
add switch hwfilter=filter-id classifier=rule-id
    action={copy|discard|forward|copy,discard|setl2qos}
```

To delete one or more packet filters from the switch, use the command:

```
delete switch hwfilter=filter-idlist
```

To delete all classifiers from a single packet filter, use the command:

```
delete switch hwfilter=filter-id classifier=all
```

To show packet filters, use the command:

```
show switch hwfilter[=filter-idlist]
```

## Access Control Lists (ACLs)

The user can configure classifiers and hardware packet filters to provide Access Control List functionality.

For example, to allow WWW servers in the 192.168.10.0 subnet to be accessed only from the 192.168.20.0 subnet:

### 1. Create a classifier to match the traffic that is to be allowed.

Create a classifier to match WWW traffic from the 192.168.20.0 subnet to the 192.168.10.0 subnet.

```
create classifier=1 ipdaddr=192.168.10.0/24
    ipsaddr=192.168.20.0/24 tcpdport=80
```

### 2. Create a hardware packet filter to allow this traffic.

```
add switch hwfilter=1 classifier=1 action=forward
    dport=all
```

### 3. Create a classifier to match the traffic that is to be denied.



Create a classifier to match all WWW traffic to the 192.168.10.0 subnet.

```
create classifier=2 ipdaddr=192.168.10.0/24 tcpdport=80
```

#### 4. Create a hardware packet filter to deny this traffic.

This filter must have a higher ID number than the allow filter because filters are processed in the order of their ID numbers.

```
add switch hwfilter=2 classifier=2 action=discard
dport=all
```

## Classifier-Based Filters with Accelerated IPv6 Traffic

The AT-8948 switch accelerates IPv6 unicast and multicast packets via an optional AT-ACC01 network processor accelerator card. When the switch receives an IPv6 packet to route, it sends the packet to the accelerator card. The card processes the packet and sends it out the correct switch port with appropriate alterations to the packet. The following table describes how you can configure the accelerator card.

| Action                                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Discard a packet                                                | It is dropped immediately.                                                                                                                                                                                                                                                                                                                                                                                   |
| Forward the packet                                              | The packet is exempt from any further filters in the Layer 3 processor of the accelerator, but you can still configure Quality of Service (QoS) controls to act on it, on the Layer 2 processor of the accelerator card and the egress switch port. See <a href="#">“Configuring QoS for Accelerated IPv6 Traffic” on page 27-34 of Chapter 27, Quality of Service (QoS)</a> for more information about QoS. |
| Remark the packet with a new DSCP and/or VLAN tag User Priority | You can then configure QoS to specify how the accelerator card and the switch handle the packet. See <a href="#">“Configuring QoS for Accelerated IPv6 Traffic” on page 27-34 of Chapter 27, Quality of Service (QoS)</a> for more information.                                                                                                                                                              |
| Send the packet to the switch CPU                               | This is a troubleshooting option to allow further analysis.                                                                                                                                                                                                                                                                                                                                                  |

For more information about IPv6 and accelerated IPv6 traffic, see [Chapter 23, Internet Protocol version 6 \(IPv6\)](#).

Accelerator hardware filters are completely independent of the switch hardware filters described in [“Classifier-Based Packet Filters” on page 7-39](#). The accelerator hardware filters are applied at the Layer 3 processor of the card (see [Chapter 27, Quality of Service \(QoS\)](#)).

## Configuring Accelerator Hardware Filters for AT-8948 Switches

### 1. Classify the accelerated IPv6 traffic.

You can classify accelerated IPv6 packets according to:

- Source/destination IPv6 address
- Layer 4 protocol (for example: TCP and UDP)
- Layer 4 source/destination ports
- IPv6 DSCP

To classify traffic, create classifiers by using the command:

```
create classifier=rule-id [ethformat={ethii-tagged|any}]
[ipdaddr={any|ipv6-add/prefix-length}] [ipsaddr={any|
ipv6-add/prefix-length}] [ipdscp={0..63|any}]
[ipprotocol={tcp|udp|icmp|ipprotocolnum|any}]
[protocol=ipv6] [tcpdport={portid|any}]
[tcpsport={portid|any}] [udpdport={portid|any}]
[udpsport={portid|any}]
```

See [Chapter 26, Generic Packet Classifier](#) for more information about the Classifier.

### 2. Create accelerator filters.

To create an accelerator filter, use the command:

```
add switch accelerator hwfilter=filter-id
classifier=rule-id action={discard|forward|mark|
sendtocpu} [newipdscp=0..63] [newpriority=0..7]
```

Accelerator filters with low filter IDs have precedence over ones with high filter IDs. Packets are matched against the order of entries in the filter table and action is taken by the accelerator filter on the first match found. For example, a packet is matched against *accelerator filter 1* before being matched against *accelerator filter 2*. Therefore, the filter configuration order is critical for the performance of classifier-based accelerator filtering.

**Other commands** To delete a specific accelerator filter or all of them, use the [delete switch accelerator hwfilter](#) command on page 7-70.

To display information about accelerator filters, including each filter's action, use the [show switch accelerator hwfilter](#) command on page 7-130.

### Interaction between switch and accelerator filters

Switch filters and accelerator filters are independent. You can configure a classifier-based switch filter on the packet's ingress port and an accelerator filter on the card.

If a switch filter on the packet's ingress port is configured to:

- **Discard** the packet, the packet is dropped immediately.
- **Forward** the packet, the packet is subject to accelerator hardware filtering. Therefore the switch filter's instruction to forward the packet can be overturned by an accelerator filter.

## Triggers

The Trigger Facility automatically runs command scripts when specific triggers are activated. When an event activates a trigger, global parameters and

parameters specific to the event are passed to the script that runs. For a full description of the Trigger Facility, see [Chapter 44, Trigger Facility](#).

The switch can generate triggers to activate scripts when a switch port goes up or down.

The following section lists events that may be specified for the **event** parameter, the parameters that may be specified as *module-specific-parameters*, and the arguments passed to the script that the trigger activates.

| <b>Event</b>             | LINKDOWN                                                                                                                                                      |           |             |                   |                                                      |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|-------------|-------------------|------------------------------------------------------|
| <b>Description</b>       | The port link specified by the <b>port</b> parameter has just gone down.                                                                                      |           |             |                   |                                                      |
| <b>Parameters</b>        | The following command parameter must be specified in the <b>create</b> and <b>set trigger</b> commands:                                                       |           |             |                   |                                                      |
|                          | <table> <tr> <th>Parameter</th><th>Description</th></tr> <tr> <td>port=<i>port</i></td><td>The port where the event activates the trigger.</td></tr> </table> | Parameter | Description | port= <i>port</i> | The port where the event activates the trigger.      |
| Parameter                | Description                                                                                                                                                   |           |             |                   |                                                      |
| port= <i>port</i>        | The port where the event activates the trigger.                                                                                                               |           |             |                   |                                                      |
| <b>Script Parameters</b> | The trigger passes the following parameter to the script:                                                                                                     |           |             |                   |                                                      |
|                          | <table> <tr> <th>Argument</th><th>Description</th></tr> <tr> <td>%1</td><td>The port number of the port that has just gone down.</td></tr> </table>           | Argument  | Description | %1                | The port number of the port that has just gone down. |
| Argument                 | Description                                                                                                                                                   |           |             |                   |                                                      |
| %1                       | The port number of the port that has just gone down.                                                                                                          |           |             |                   |                                                      |
| <b>Event</b>             | LINKUP                                                                                                                                                        |           |             |                   |                                                      |
| <b>Description</b>       | The port link specified by the <b>port</b> parameter has just come up.                                                                                        |           |             |                   |                                                      |
| <b>Parameters</b>        | The following command parameter must be specified in the <b>create</b> and <b>set trigger</b> commands:                                                       |           |             |                   |                                                      |
|                          | <table> <tr> <th>Parameter</th><th>Description</th></tr> <tr> <td>port=<i>port</i></td><td>The port where the event activates the trigger.</td></tr> </table> | Parameter | Description | port= <i>port</i> | The port where the event activates the trigger.      |
| Parameter                | Description                                                                                                                                                   |           |             |                   |                                                      |
| port= <i>port</i>        | The port where the event activates the trigger.                                                                                                               |           |             |                   |                                                      |
| <b>Script Parameters</b> | The trigger passes the following parameter to the script:                                                                                                     |           |             |                   |                                                      |
|                          | <table> <tr> <th>Argument</th><th>Description</th></tr> <tr> <td>%1</td><td>The port number of the port that has just come up.</td></tr> </table>             | Argument  | Description | %1                | The port number of the port that has just come up.   |
| Argument                 | Description                                                                                                                                                   |           |             |                   |                                                      |
| %1                       | The port number of the port that has just come up.                                                                                                            |           |             |                   |                                                      |

To create or modify a switch trigger, use the commands:

```
create trigger=trigger-id module=switch port=port
  event={linkdown|linkup} [after=hh:mm] [before=hh:mm]
  [{date=date|days=day-list}] [name=name] [repeat={yes|no|
  once|forever|count}] [script=filename...] [state={enabled|
  disabled}] [test={yes|no|on|off|true|false}]

set trigger=trigger-id [port=port] [after=hh:mm]
  [before=hh:mm] [{date=date|days=day-list}] [name=name]
  [repeat={yes|no|once|forever|count}] [test={yes|no|on|
  off|true|false}]
```

## Configuration Examples

This section shows the following examples:

- [Port-Based VLAN with Untagged Ports](#)
- [VLAN with Tagged Ports](#)

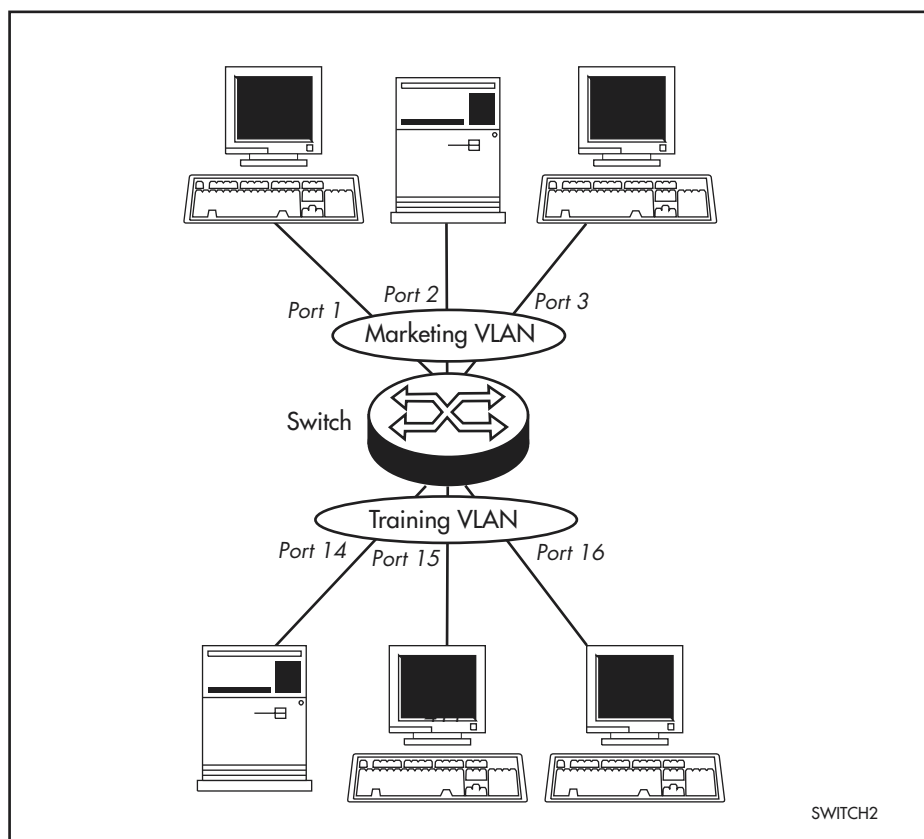
### ■ Subnet-Based VLAN

Examples assume that the switch configuration begins from factory default settings. Note that routing, required for communication between the VLANs is not shown in these examples.

## Port-Based VLAN with Untagged Ports

Two VLANs using untagged ports are shown in [Figure 7-7](#). Ports 1-3 belong to one broadcast domain, the marketing VLAN, and ports 14-16 belong to another broadcast domain, the training VLAN. The switch acts as two separate bridges: one that forwards between the ports belonging to the *marketing* VLAN, and a second one that forwards between the ports belonging to the *training* VLAN. A frame is assigned to a VLAN based on the switch port on which the frame arrives. Devices on ports 2 and 14 communicate with each other by using IP routing functions only.

Figure 7-7: Port-based VLANs using untagged ports



The parameters used to configure this example are shown in [Table 7-4](#). Since there is only one switch and no loops in this topology, the Spanning Tree Protocol (STP) is not needed.

Table 7-4: Parameters for port-based VLAN example

| VLAN name | VLAN ID | Ports      |
|-----------|---------|------------|
| Marketing | VID=2   | PORT 1-3   |
| Training  | VID=3   | PORT 14-16 |

### Configure the switch

#### I. Create VLANs.

Create the two VLANs using the following commands on the switch:

```
create vlan=marketing vid=2
create vlan=training vid=3
```

### 2. Add ports to VLANs.

Add the ports to these VLANs on the switch by using the following commands:

```
add vlan=marketing port=1-3
```

```
add vlan=training port=4-6
```

Check the VLAN configuration by using the command:

```
show vlan
```

### 3. Check the switch.

Check that the switch is switching across the ports. Traffic on the switch can be monitored using the command:

```
show switch counter
```

## VLAN with Tagged Ports

A network that must be configured with VLAN tagging since the VLAN aware server on port 2 on Switch A belongs to both the admin VLAN and the marketing VLAN is shown in [Figure 7-8](#). Using VLAN tags, port 26 on Switch A and port 25 on Switch B belong to both the marketing VLAN and the training VLAN so that devices on both VLANs can use this uplink to communicate with other devices in the same VLAN on the other switch. There are no loops in this topology so STP is not needed. The parameters used to configure this example are shown in [Table 7-5](#).

Figure 7-8: VLANs using tagged ports

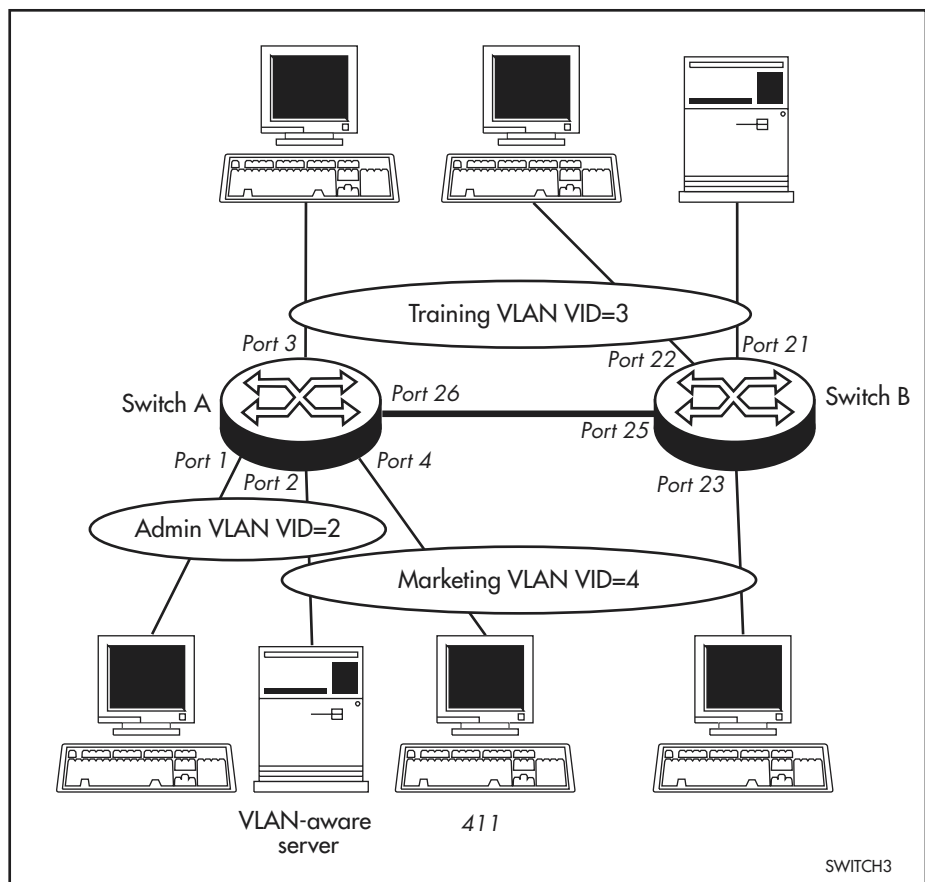


Table 7-5: Configuration parameters for VLANs

| VLAN name | VID   | Switch A     |                | Switch B     |                |
|-----------|-------|--------------|----------------|--------------|----------------|
|           |       | Tagged ports | Untagged ports | Tagged ports | Untagged ports |
| Admin     | VID=2 | PORT 2       | PORT 1         |              |                |
| Training  | VID=3 | PORT 26      | PORT 3         | PORT 25      | PORT 21,22     |
| Marketing | VID=4 | PORT 2,26    | PORT 4         | PORT 25      | PORT 23        |

### Configure Switch A

#### 1. Create VLANs.

Create the three VLANs using the following commands on the switch:

```
create vlan=admin vid=2
create vlan=training vid=3
create vlan=marketing vid=4
```

#### 2. Add ports to VLANs.

Add the ports to these VLANs on the switch by using the following commands:

```
add vlan=admin port=2 frame=tagged
add vlan=admin port=1 frame=untagged
add vlan=training port=26 frame=tagged
add vlan=training port=3 frame=untagged
add vlan=marketing port=2,26 frame=tagged
add vlan=marketing port=4 frame=untagged
```

Check the VLAN configuration by using the command:

```
show vlan
```

### Configure Switch B

#### 1. Create VLANs.

Create the two VLANs using the following commands on the switch:

```
create vlan=training vid=3
create vlan=marketing vid=4
```

#### 2. Add ports to VLANs.

Add the ports to these VLANs on the switch by using the following commands:

```
add vlan=training port=5 frame=tagged
add vlan=training port=1,2 frame=untagged
add vlan=marketing port=5 frame=tagged
add vlan=marketing port=3 frame=untagged
```

Check the VLAN configuration by using the command:

```
show vlan
```

### Check

Check that the switch is switching across the ports. Traffic on Switch A can be monitored using the command:

```
show switch counter=1-4,26
```

Traffic on Switch B can be monitored using the command:

```
show switch counter=21-23,25
```

## Subnet-Based VLAN

How data link layer VLANs automatically map to subnets in the network layer is shown in [Figure 7-9](#). The IP address of the device automatically places it in a VLAN appropriate for the subnet. If traffic is not IP traffic, then protocol or port-based classification rules are used. The parameters used to configure this example are shown in [Table 7-6 on page 7-48](#).

Figure 7-9: Subnet-based VLAN example

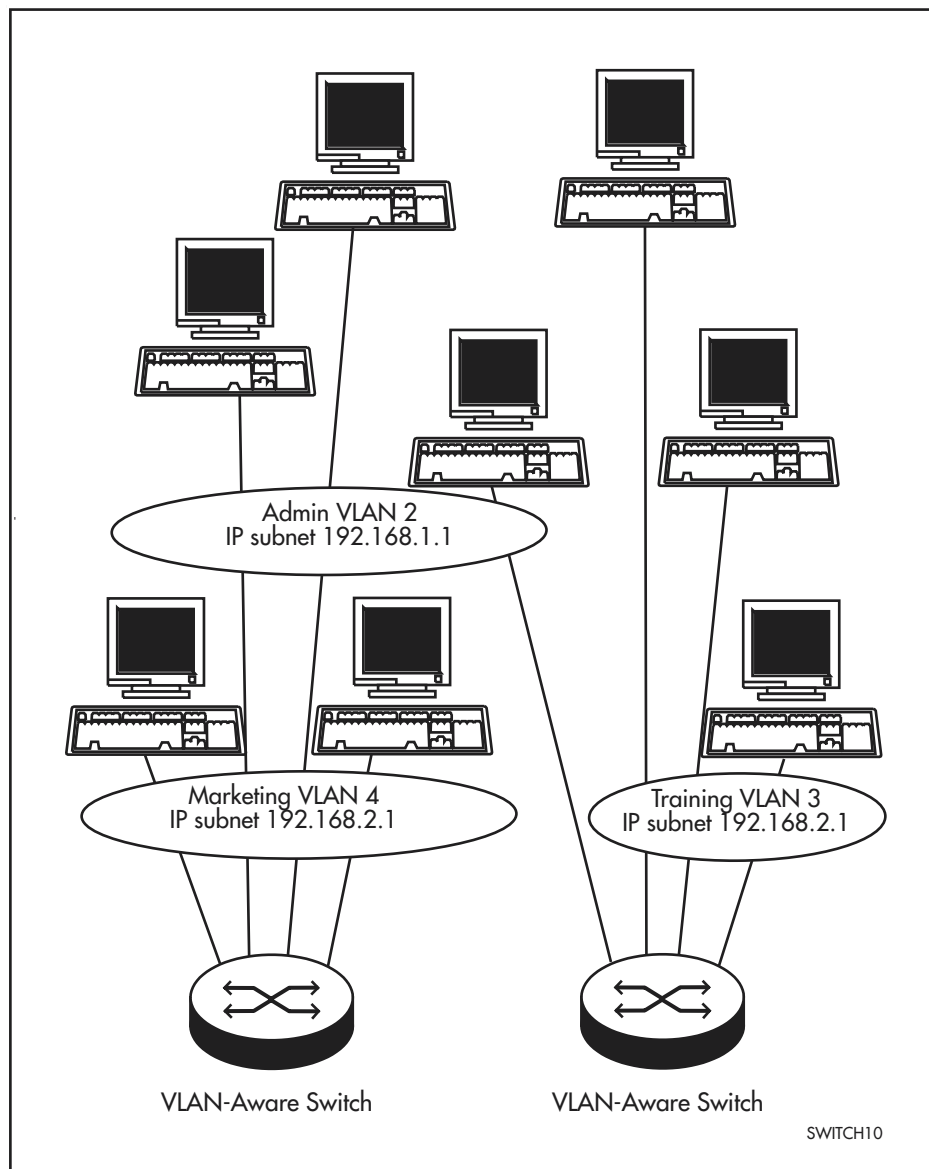


Table 7-6: Configuration parameters for VLANs with subnet associations

| Switch A  |       | Switch B     |                |
|-----------|-------|--------------|----------------|
| VLAN name | VID   | Tagged ports | Untagged ports |
| Admin     | VID=2 | PORT 1       |                |
| Training  | VID=3 | PORT 3       | PORT 21,22     |
| Marketing | VID=4 | PORT 4       | PORT 23        |

## Configure Switch A

### 1. Create VLANs.

Create the three VLANs using the following commands on the switch:

```
create vlan=admin vid=2 type=subnet
create vlan=training vid=3 type=subnet
create vlan=marketing vid=4 type=subnet
```

### 2. Add subnets to VLANs.

Add the subnets to these VLANs on the switch by using the following commands:

```
add vlan=admin subnet=192.168.1.1
add vlan=training subnet=192.168.2.1
add vlan=marketing subnet=192.168.3.1
```

### 3. Add ports to VLANs.

Add ports to these VLANs on the switch by using the following commands:

```
add vlan=admin port=2 subnet=192.168.1.1
add vlan=training port=3 subnet=192.168.2.1
add vlan=marketing port=4 subnet=192.168.3.1
```

### 4. Check the VLAN configuration by using the command:

```
show vlan
```

## Configure Switch B

### 1. Create VLANs.

Create the three VLANs using the following commands on the switch:

```
create vlan=admin vid=2 type=subnet
create vlan=training vid=3 type=subnet
create vlan=marketing vid=4 type=subnet
```

### 2. Add subnets to VLANs.

Add the subnets to these VLANs on the switch by using the following commands:

```
add vlan=admin subnet=192.168.1.1
add vlan=training subnet=192.168.2.1
add vlan=marketing subnet=192.168.3.1
```



**3. Add ports to VLANs.**

Add ports to these VLANs on the switch by using the following commands:

```
add vlan=admin port=2 subnet=192.168.1.1
add vlan=training port=3 subnet=192.168.2.1
add vlan=marketing port=4 subnet=192.168.3.1
```

**Check**

Check that the switch is switching across the ports. Traffic on Switch A can be monitored using the command:

```
show switch port=1-4,26 counter
```

Traffic on Switch B can be monitored using the command:

```
show switch port=21-23,25 counter
```

## Command Reference

---

This section describes the commands available to configure and manage switching functions on the switch.

The shortest valid command is denoted by capital letters in the Syntax section. See [“Conventions” on page -xlix of , About this Software Reference](#) in the front of this manual for details of the conventions used to describe command syntax. See [Appendix A, Messages](#) for a complete list of messages and their meanings.

### activate switch port

---

**Syntax** ACTivate SWItch POrt={*port-list*|ALL} [AUTOnegotiate] [LOCK]

where *port-list* is a port number, range (specified as *n-m*), or comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.

**Description** This command activates autonegotiation of port speed and duplex mode for a port or a group of ports.

The **port** parameter specifies the port for which autonegotiation is to be activated. Only ports in the list that are set to autonegotiate are affected by this command. Ports are not modified that have a fixed speed setting or that belong to a trunk group.

A port that has been added to LACP autonegotiates until it actively becomes part of an aggregated link (i.e. trunked), when it then operates at the speed of the aggregated link.

The **autonegotiate** parameter specifies that the port is to activate the autonegotiation process. The port begins to autonegotiate link speed and duplex mode.

The **lock** parameter manually locks the switch port before it reaches its learning limit so that no new addresses are automatically learned. The **learn** parameter for the port is set to the current number of learned MAC addresses.

**Examples** To activate autonegotiation on ports 1-8 and port 10, use the command:

```
act swi po=1-8,10 auto
```

**Related Commands** [set switch port](#)  
[show switch port](#)

## add lacp port

---

**Syntax** ADD LACP Port=[*{port-list|ALL}*] [Adminkey=*key*]  
[Priority=*priority*] [Mode={Active|Passive}]  
[Periodic={Fast|Slow}]

where:

- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered port, including uplink ports.
- *key* is an integer from 0 to 65535
- *priority* is an integer from 0 to 65535

**Description** This command adds a port to LACP's control thus enabling LACP to put it into an aggregated link. By default, ports are added in the active mode. If a port is added in the active mode, and its link's requirements for trunking are met, then the port and its associated link are automatically aggregated without further configuration. The same situation applies for a port configured in passive mode but whose link connects to a remote port configured in active mode. To run LACP the port must be operating in the full duplex mode.

The **port** parameter specifies the ports whose parameters are to be modified. Where none of the ports specified are presently managed by LACP, the command takes effect if it can be applied to all the specified ports. Where some of the ports specified are already managed by LACP, and additional ports are added (by specifying **all**, for example), then the LACP managed ports have their Key and other parameters changed, and the command succeeds on all the specified ports.

In the following descriptions, references to an individual port refers to all ports selected by the **port** parameter.

The **adminkey** parameter specifies the Admin LACP port key. This affects the LACP port key that is generated but does not determine its value. You can use this parameter to prevent ports from being aggregated when they might otherwise form a trunk. By default all ports that can be aggregated are given the same LACP port key. The default for **adminkey** is 1.

The **priority** parameter specifies the *LACP port priority*. The priority assigned is used where the number of physical links connecting two devices is greater than the number that can be aggregated. The priority entered is then used to determine which ports are selected for aggregation. The default of 32,768 (0 being the highest priority) is applied to all ports.

Where the port priority is the same, the port number governs which ports are selected. The lower the port number, the higher its priority. Excess ports are put into a standby mode, in which they are effectively disabled. They will remain in this state unless required to replace inoperative links within their associated aggregated group.

The **mode** parameter specifies whether the port runs in LACP *passive* or *active* mode. A port in passive mode begins sending LACPDU's in response to a received LACPDU; whereas, a port in active mode always sends LACPDU's at regular intervals specified by the **periodic** parameter.

The **periodic** parameter specifies the requested rate that the LACP port receives LACPDU *update messages* from its partner port. A port in fast mode receives one LACPDU every second; in slow mode, a port receives one every thirty seconds.

**Examples** To add ports 3 and 5 to LACP, use the command:

```
add lacp po=3,5
```

**Related Commands**

- [delete lacp port](#)
- [disable lacp](#)
- [disable stp](#)
- [set lacp port](#)
- [show lacp port](#)

## add switch accelerator hwfilter

**Syntax** `ADD SWITch ACCELerator HWFilter=filter-id  
CLASSifier=rule-id Action={DIScard|FORward|Mark|  
SENDtocpu} [NEWIPDscp=0..63] [NEWPriority=0..7]`

where:

- *filter-id* is a decimal number from 1 to 2000
- *rule-id* is the number of the classifier to attach to this hardware filter

**Description** This command adds a hardware-based packet filter to the IPv6 accelerator card on AT-8948 switches. The amount of available memory in the IPv6 accelerator card determines the number of hardware-based packet filters you can add. Hardware-based packet filters defined for the IPv6 accelerator card are separate from those defined for the base switch.

The **hwfilter** parameter specifies a unique numerical identifier for the hardware-based packet filter. Hardware-based packet filters with lower filter IDs have a greater precedence than those with higher ones. Therefore, we recommend that you give specific filters lower IDs and give general filters higher ones.

The **classifier** parameter specifies the rule ID of an existing packet matching rule (a classifier). To create a packet matching rule use the [create classifier command on page 26-4 of Chapter 26, Generic Packet Classifier](#).

The **action** parameter specifies what action the classifier entry takes with packets that match the classifier. If **discard** is specified, packets that match the classifier are discarded. If **forward** is specified, packets that match the classifier are forwarded. If **sendtocpu** is specified, packets that match the classifier are sent to the CPU. If **mark** is specified, you must specify a value for one or both of the **newpriority** and **newipdscp** parameters. If **mark** and **newpriority** are specified, the VLAN Tag User Priority of the packet is replaced with the value specified for the **newpriority** parameter. If **mark** and **newipdscp** are specified, the IPv6 DSCP field of the packet is replaced with the value specified for the **newipdscp** parameter.

The **newpriority** parameter specifies the new 802.1q priority value assigned to the packet if an action of **mark** action is specified. If **newpriority** is specified a value must be supplied; there is no default.

The **newipdscp** parameter specifies the new IPv6 DSCP value assigned to the packet if an action of **mark** action is specified. If **newipdscp** is specified a value must be supplied; there is no default.

**Example** To add a hardware-based packet filter to the IPv6 accelerator card based upon packet matching rule 2 to discard traffic that matches the classifiers, and put the filter in filter position 1, use the command:

```
add swi accel hwf=1 class=2 ac=dis
```

To add a hardware-based packet filter to the IPv6 accelerator card based upon packet matching rule 2 to change the DSCP of matching traffic to 8, and put the filter in filter position 1, use the command:

```
add swi accel hwf=1 class=2 ac=ma newipd=8
```

**Related Commands**

- [add switch hwfilter](#)
- [delete switch accelerator hwfilter](#)
- [show switch accelerator](#)
- [show switch accelerator counter](#)
- [show switch accelerator hwfilter](#)

## add switch filter action

---

**Syntax** ADD SWITch FILTER ACTion={FORward|DIScard}  
DESTaddress=*macadd* PORT=*port* [ENTry=*entry*] [LEARn]  
[VLAN={*vlan-name*|1..4094}]

where:

- *entry* is a filter entry number from 0 to n+1 where n is the highest filter entry currently defined in the permanent forwarding database. The permanent forwarding database has a maximum of 320 entries that range from 0 to 319. Each port has its own permanent forwarding database.
- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.
- *port* is the number of the switch port or uplink port to which this filter applies.
- *macadd* is an Ethernet six-octet MAC address expressed as six pairs of hexadecimal digits delimited by hyphens.

**Description** This command adds a single static filter entry to the permanent forwarding database for a specific port. If the static entry matches an existing dynamic entry that was learned by the switch (a match means that the **destaddress** and VLAN parameters are the same for both entries), the static filter overwrites the existing dynamic learned entry. Received frames that match the static filter entry are forwarded to the specified port with an action of either **forward** or **discard**.

The **action** parameter specifies the outcome of the forwarding process for the frame. If **forward** is specified, the frame is transmitted on the given port. If **discard** is specified, all frames that have the specified destination MAC address and belong to the specified VLAN will be dropped, irrespective of the destination port that the frame will be transmitted from. The frame does not need to match the port specified for the **discard** action to occur.

The **destaddress** parameter specifies the value to be matched against the destination MAC address from frames being filtered. The destination MAC address must be an individual MAC address, and cannot be the MAC address of the switch.

The **port** parameter specifies the outbound port over which a frame matching this filter entry is discarded or forwarded. Whether the ports are tagged ports or untagged ports is determined by the **vlan** parameter. If the **port** parameter specifies tagged ports, then the **vlan** parameter is required.

The **entry** parameter specifies where in the permanent forwarding database the new entry is added for the specified port. **entry** cannot be set greater than n+1 where n is the highest filter entry currently defined. If **entry** is not specified, the new entry is appended to the bottom of the permanent forwarding database: the default is n+1 where n is the highest filter entry currently defined. Static and dynamic entries in the forwarding database are kept in sorted order determined by their VLAN Identifier (VID) and MAC address. Therefore the **entry** parameter does not affect the order of the filters in the forwarding database. The order in which filter entries are displayed by the **show switch filter** command depends on the **entry** parameter.

The **learn** parameter specifies if the filter being added should be counted and used as a learned MAC address for intrusion detection. Learned filters are not totally static, and can be lost if the learning process is stopped by setting the **learn** parameter to zero.

The **vlan** parameter specifies the VID to which the filter entry is associated. The **vlan** parameter is required if the **port** parameter specifies tagged ports. If the **port** parameter specifies untagged ports, the **vlan** parameter is not required, and defaults to the VID of the VLAN for which the ports are untagged. Therefore, if the **vlan** parameter is not specified, the ports are treated as untagged ports.

The switch automatically deletes static filter entries for a port when the port is deleted from a specific VLAN.

**Examples** To forward all frames destined for MAC address 00-00-cd-12-34-56 on the VLAN to which port 3 is an untagged port, use the command:

```
add swi fil dest=00-00-cd-12-34-56 ac=for po=3
```

To discard all frames destined for MAC address 00-00-cd-12-34-56 on port 4 in VLAN 4, use the command:

```
add swi fil dest=00-00-cd-12-34-56 po=4 ac=dis vlan=4
```

**Related Commands** [delete switch filter](#)  
[show switch filter](#)

## add switch hwfilter

---

**Syntax** ADD SWITCh HWFilter=*filter-id* CLASSifier=*rule-id*  
 Action={COPY|DIScard|FORward|COPY,DIScard|SETL2QOS}  
 [L2QOSqueue=0..7] [PRIOrity=0..7] [BANDwidthclass=1..3]

where:

- *filter-id* is a decimal number from 1 to 1024.
- *rule-id* is a integer from 1 to 9999.

**Description** This command adds a hardware-based packet filter to the switch. The number of filters that can be added is determined by the amount of available space in the hardware-based packet classification tables. Hardware packet filters apply to packets received on any port, but a classifier within a filter can match on source VLAN.

The **hwfilter** parameter specifies a unique numerical identifier for the hardware-based packet filter. Packet filters with lower filter IDs have a greater precedence than those with higher ones. Therefore, we recommend that you give specific filters lower IDs and give general filters higher ones.

The **classifier** parameter specifies the rule-id of an existing packet matching rule or classifier. A packet matching rule is created using the [create classifier command on page 26-4 of Chapter 26, Generic Packet Classifier](#).

The **action** parameter specifies what action the classifier entry takes with packets that match the classifier. If **discard** is specified, then packets matching the classifier are discarded. If **forward** is specified, then packets matching the classifier are forwarded. If **copy** is specified, then a copy of packets matching the classifier are sent to the CPU and the packets are forwarded. If **copy, discard** is specified, then a copy of packets matching the classifier are sent to the CPU and the packets are discarded. If **setl2qos** is specified, then packets matching the classifier have their bandwidth class (or drop precedence), queue, and 802.1p user priority modified to the values specified by the **bandwidthclass**, **l2qosqueue**, and **priority** parameters. The action **setl2qos** cannot be specified at the same time as **drop**.

The **l2qosqueue** parameter specifies the queue to send packets to that match the classifier. The default is 0.

The **priority** parameter specifies the 802.1p user priority to remark packets with that match the classifier. The default is 0.

The **bandwidthclass** parameter specifies the bandwidth class (drop precedence) to assign packets to that match the classifier. The default is 1.

**Examples** To add a hardware-based packet filter based upon packet matching rule 2 to discard traffic that matches the classifiers, use the command:

```
add swi hwf=1 class=2 ac=dis
```

**Related Commands** [create classifier](#)  
[delete switch hwfilter](#)  
[show switch hwfilter](#)



## add switch trunk

**Syntax** ADD SWItch TRunk=*trunk* POrt=*port-list*

where:

- *trunk* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. Do not name a trunk *lacp* because the switch automatically adds this prefix when it creates an LACP trunk group (or aggregated link).
- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.

**Description** This command adds ports to an existing trunk group on the switch. Port trunking **must** be configured on both ends of a link or network loops may result. The switch supports static 802.3ad link aggregation, and port trunking is also called *link aggregation* (for details, see [“Link Aggregation” on page 7-12](#)).

Ports in a trunk group must have the same VLAN configuration and it is not possible to have more than one uplink port in a VLAN, unless the uplink ports are members of the same trunk group. Therefore, if private VLANs are used, ensure that when adding ports to a trunk group that none of the ports is an uplink port. For more information about the private VLAN feature, see [“Private VLANs” on page 7-30](#).

The **trunk** parameter specifies a unique name for the trunk group. The name is not case sensitive although the case you enter is preserved for display purposes.

The **port** parameter specifies the switch ports to be added to the trunk group. Ports specified must not be in another trunk group, and must have the same VLAN configuration. They cannot include the switch’s mirroring port. The following table explains ports in trunking groups.

| Switch    | Max No. of Trunks | Max No. of Ports per Trunk |
|-----------|-------------------|----------------------------|
| AT-8948   | 7                 | 4                          |
| x900-48   | 7                 | 8                          |
| AT-9900   | 12                | 4                          |
| AT-9924Ts | 12                | 4                          |
| x900-24X  | 12                | 8                          |

When you add a port to a trunk group, the switch saves the port’s current speed and duplex mode settings and sets the port to autonegotiate to the speed of the trunk group and full duplex mode. If the port does not support auto-negotiation, the switch sets the port to the fixed speed of the trunk group and full duplex mode. When you remove a port from a trunk group, the switch restores the port’s speed and duplex mode settings to the values they had before the port was added to the trunk group.

**Example** To add ports 5 and 6 to trunk group Trunk1, use the command:

```
add swi tr=Trunk1 po=5,6
```

**Related Commands**

- [create switch trunk](#)
- [delete switch trunk](#)
- [destroy switch trunk](#)
- [set switch trunk](#)
- [show switch trunk](#)

## add vlan port

**Syntax** ADD VLAN={*vlan-name*|1..4094} Port={*port-list*|ALL}  
                   [FRame={TAGged|UNTAGged}] [UPLink]

ADD VLAN={*vlan-name*|1..4094} Port={*port-list*|ALL}  
                   SUBNET={*ipadd*|ALL} [UPLink]

ADD VLAN={*vlan-name*|1..4094} Port={*port-list*|ALL}  
                   PROTOcol={*protocol-type*|*index-list*|ALL} [UPLink]

ADD VLAN={*vlan-name*|1..4094} Port={*port-list*}  
                   NEStedtype={CORE|CUStomer}

ADD VLAN={*vlan-name*|1..4094} Port={*port-list*|ALL}  
                   SUBNET={*ipadd*|ALL} NEStedtype={CORE|CUStomer}

ADD VLAN={*vlan-name*|1..4094} Port={*port-list*}  
                   PROTOcol={*protocol-type*|*index-list*|ALL}  
                   NEStedtype={CORE|CUStomer}

where:

- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.
- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port.
- *ipadd* is an IP address in dotted decimal notation.
- *protocol-type* is either a valid protocol number or a recognised protocol name. A protocol number can be 1 byte for SAP, 2 bytes for ETHII, or 5 bytes for an 802.3/802.2 SNAP packet. The protocol-type is specified as a hexadecimal number, which begins with "0x".
- *index-list* is a single index number or a group as either a comma-separated list, range (specified as *n-m*), or a combination of the two. Index numbers start at 0.

**Description** This command adds ports to the specified VLAN.

A port cannot be a member of both a private VLAN and a non-private VLAN. See [“Private VLANs” on page 7-30](#) for more information about configuring private VLANs.

All ports that are members of a specific trunk group must have the same VLAN configuration. Therefore, all ports in a trunk group must be specified if a port that is a member of that trunk group is to be added to a VLAN.

The ports must belong to only one STP after being added to the VLAN. If as a result of the port addition, ports are moved from one STP to another STP, the two affected STPs are initialised if they are currently enabled. Any previously disabled ports are STP enabled.

If the ports transmit untagged packets for the VLAN, this command can associate an IP subnet and protocol with the specified ports. Also, this command can associate the Port VLAN classification rule of the VLAN with the ports if the ports transmit untagged packets for the VLAN. When a port is

added to a VLAN for the first time, the tagged egress status of the port is set for the VLAN. To change the tagged status of the port, delete the port completely from the VLAN and re-add the port to the VLAN specifying the **frame** parameter.

The **vlan** parameter specifies the name or numerical VLAN Identifier (VID) of the VLAN. The name is not case sensitive. However, the case is preserved for display purposes. The VLAN must already exist. By default, all ports belong to the default VLAN which has a numerical VLAN Identifier (VID) of 1.

The **port** parameter specifies the ports to add to the selected VLAN. The command will fail unless successful on all ports. The following rules apply when adding ports to VLANs:

- You cannot add mirror ports to a VLAN.
- All ports in a trunk group must belong to the same VLAN and spanning tree.
- You cannot delete trunked ports from the default VLAN. For example, you cannot add a trunk member port as an untagged port of another VLAN, because this would result in a trunk with mixed VLAN membership.

If the optional **subnet** or **protocol** parameters are not specified the ports are associated with the Port VLAN classification rule of the VLAN. If the ports were not members of the VLAN prior to this command being processed and the **frame** parameter does not specify **tagged**, then the ports are implicitly deleted from the default VLAN and added to this VLAN before the association to the Port VLAN classification rule of this VLAN is made.

If the VLAN is a private VLAN and the **uplink** parameter is not specified, then the ports are added as private ports. All traffic received on a private port is sent to a predefined uplink port. Private ports cannot be added to a private VLAN until the VLAN has an uplink port or trunk group added to it. The port cannot already be in a non-private VLAN, and the port must not be an uplink port in other VLANs. If the port is already in a private VLAN, then the two VLANs must have the same uplink port. See [“Private VLANs” on page 7-30](#) for information about configuring private VLANs.

The **frame** parameter specifies whether a VLAN tag header is included in each frame transmitted on the specified ports. If **tagged** is specified, a VLAN tag is added to frames prior to transmission and the port is called a tagged port for this VLAN. If **untagged** is specified, the frame is transmitted without a VLAN tag and the port is called an untagged port for this VLAN. A port can be **untagged** for one and only one of the VLANs to which it belongs, or for none of the VLANs to which it belongs. A port can have the **frame** parameter set to **tagged** for zero or more VLANs to which it belongs. It is not possible to add an untagged port to a VLAN if the port is already present in any other VLAN except the default VLAN. If the port is an untagged member of the default VLAN, adding it untagged to another VLAN deletes it from the default VLAN. The default setting is **untagged**.

The **frame** parameter can only be specified the first time a port is added to a VLAN. This determines whether the port transmits untagged or VLAN-tagged packets for the VLAN. Subsequent **add vlan port** commands that specify the same port and VLAN cannot specify the **frame** parameter.

The **subnet** parameter is used to associate an IP subnet with a port if the port transmits untagged packets for the VLAN. If the port is not a member of the VLAN, and the **frame** parameter does not specify **tagged**, the port is added to the VLAN and the association is made afterwards. If the **subnet** parameter

specifies **all** or no value, all previously added IP subnets are associated with the port. See the [add vlan subnet command on page 7-64](#) for more details.

The **protocol** parameter is used to associate one or more protocols with a port if the port transmits untagged packets for the VLAN. If the port is not a member of the VLAN, and the **frame** parameter does not specify **tagged**, the port is added to the VLAN and the association is made afterwards. If the **protocol** parameter specifies an index number, the index number represents a protocol previously added to the VLAN. If the **protocol** parameter specifies **all** or no value, then all previously added protocols are associated with the port. See the [add vlan protocol command on page 7-62](#) for more details.

Combinations of the **subnet** or **protocol** parameters cannot be specified on the command line at the same time. If **tagged** is specified for the **frame** parameter, then the optional **subnet** or **protocol** parameters cannot be specified.

A port that transmits untagged packets for a VLAN must be associated with at least one IP subnet or protocol belonging to the VLAN, or be associated with the Port VLAN classification rule of the VLAN.

The **nestedtype** parameter specifies the type of nested port. If **core** is specified, the port is connected to a service provider network. A Core port has a Customer-ID (CID) embedded in transmitted packets except for control packets. If **customer** is specified, the port is connected to a customer or private network. A Customer port does not have a CID embedded in transmitted packets. Nested ports can be added to the VLAN only when the VLAN is first created as a nested VLAN with the [create vlan command on page 7-68](#).

The **uplink** parameter specifies that the ports are to be added to the VLAN as uplink ports, and is valid only for private VLANs. If more than one port is specified, then they must be a trunked group. Each private VLAN can have only one uplink. A port that belongs to a private VLAN cannot also belong to any non-private VLAN. The port cannot be a private port in a VLAN. No other ports in the VLAN can already be associated with a different uplink in another private VLAN. See [“Private VLANs” on page 7-30](#) for information about configuring private VLANs.

**Examples** To add port 4 to the port-based Marketing VLAN as an untagged port, use either of the commands:

```
add vlan=marketing po=4
add vlan=marketing po=4 fra=untag
```

On a switch with 26 ports, to add all ports to the Sales VLAN as untagged ports, use either of the commands:

```
add vlan=sales po=all
add vlan=sales po=1-26
```

To add ports 1-4 to the Sales VLAN for the first time, so that they transmit untagged packets for the Sales VLAN, and to associate them with all IP subnets previously added to the VLAN, use:

```
add vlan=sales po=1-4 subnet=all
```

To add ports 1-4 to the Sales VLAN for the first time, so that they transmit untagged packets for the sales VLAN and to associate them with all IP subnets previously added to the VLAN, use either of the commands:

```
add vlan=sales po=1-4 subnet=192.133.23.0 mask=255.255.255.0
add vlan=sales po=1-4 subnet=192.133.23.0
```

To associate ports 1-4, which were previously added to the Sales VLAN with the specification `frame=untagged`, with all IP subnets previously added to the *sales* VLAN, use either of the commands:

```
add vlan=sales po=1-4 subnet
```

```
add vlan=sales po=1-4 subnet=all
```

To add ports 1-4 to the Sales VLAN so that they transmit VLAN-tagged packets for the sales VLAN, use:

```
add vlan=sales po=1-4 fra=tag
```

**Related Commands**

- [add vlan protocol](#)
- [add vlan subnet](#)
- [delete vlan port](#)
- [set vlan port](#)
- [show vlan](#)

## add vlan protocol

**Syntax** ADD VLAN={*vlan-name*|1..4094} PROTOcol=*protocol-type*

where:

- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.
- *protocol-type* is either a valid protocol number or a recognised protocol name. A protocol number is a hexadecimal number which can be either 1 byte for SAP, 2 bytes for ETHII or 5 bytes for an 802.3/802.2 SNAP type packet. Hexadecimal numbers must begin with "0x".

**Description** This command adds a protocol to an IP subnet-based or protocol-based VLAN. If the protocol is to be used to map untagged packets received by the port to the VLAN, then each protocol must be associated with a port in the VLAN. After adding the protocol to the VLAN, the protocol can be referred to by its index number in subsequent VLAN commands. The index number for a particular protocol may change as protocols are added to, or deleted from, the VLAN. The index numbers form a contiguous range.

The **vlan** parameter specifies the name or numerical VLAN Identifier (VID) of the IP subnet-based or protocol-based VLAN. The name is not case sensitive, however, the case is preserved for display purposes. The VLAN specified must exist.

The **protocol** parameter specifies an Ethernet protocol. The VLAN module provides a predefined list of common protocols ([Table 7-7](#)).

Table 7-7: Predefined protocol types implemented by the VLAN module

| Protocol Name    | Protocol Number | Encapsulation |
|------------------|-----------------|---------------|
| SNA Path Control | 0x04            | SAP           |
| PROWAY-LAN       | 0x0E            | SAP           |
| EIA-RS           | 0x4E            | SAP           |
| PROWAY           | 0x8E            | SAP           |
| IPX 802.2        | 0xE0            | SAP           |
| NetBEUI          | 0xF0            | SAP           |
| ISO CLNS IS      | 0xFE            | SAP           |
| IP               | 0x0800          | EthII         |
| X.75 Internet    | 0x0801          | EthII         |
| NBS Internet     | 0x0802          | EthII         |
| ECMA Internet    | 0x0803          | EthII         |
| Chaosnet         | 0x0804          | EthII         |
| X.25 Level 3     | 0x0805          | EthII         |
| ARP              | 0x0806          | EthII         |
| XNS Compat       | 0x0807          | EthII         |
| Banyan Systems   | 0x0BAD          | EthII         |
| BBN Simnet       | 0x5208          | EthII         |
| DEC MOP Dump/Ld  | 0x6001          | EthII         |

Table 7-7: Predefined protocol types implemented by the VLAN module (cont)

| Protocol Name    | Protocol Number | Encapsulation     |
|------------------|-----------------|-------------------|
| DEC MOP Rem Cons | 0x6002          | EthII             |
| DEC DECNET       | 0x6003          | EthII             |
| DEC LAT          | 0x6004          | EthII             |
| DEC Diagnostic   | 0x6005          | EthII             |
| DEC Customer     | 0x6006          | EthII             |
| DEC LAVC         | 0x6007          | EthII             |
| RARP             | 0x8035          | EthII             |
| DEC LANBridge    | 0x8038          | EthII             |
| DEC Encryption   | 0x803D          | EthII             |
| AppleTalk        | 0x809B          | EthII             |
| IBM SNA          | 0x80D5          | EthII             |
| IPX              | 0x8137          | EthII             |
| AppleTalk AARP   | 0x80F3          | EthII             |
| SNMP             | 0x814C          | EthII             |
| IPv6             | 0x86DD          | EthII             |
| IPX 802.3        | 0xFFFF          | NetWare 802.3 Raw |
| ETHERTALK2       | 0x080007809B    | SNAP              |
| ETHERTALK 2 AARP | 0x00000080F3    | SNAP              |
| IPX SNAP         | 0x0000008137    | SNAP              |

The **add vlan port** command should be used to associate the specified protocol with one or more ports in the VLAN.

**Examples** To add the IP protocol to the Marketing protocol-based VLAN, use either of the following commands:

```
add vlan=marketing prot=ip
add vlan=marketing prot=0X0800
```

To add the IPX protocol for ETHERNET-II packets to the Sales IP subnet-based VLAN, use either of the following commands:

```
add vlan=sales prot=ipx
add vlan=sales prot=0x8137
```

**Related Commands**

- [add vlan port](#)
- [delete vlan port](#)
- [delete vlan protocol](#)
- [show vlan](#)

## add vlan subnet

---

**Syntax** ADD VLAN={*vlan-name*|1..4094} SUBNET=*ipadd* [MASK=*ipadd*]

where:

- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.
- *ipadd* is an IP address in dotted decimal notation.

**Description** This command adds an IP subnet to an IP subnet-based VLAN. Each IP subnet must be associated with a port in the VLAN if the IP subnet is to be used to map untagged packets received by the port to the VLAN. The IP subnets added to a VLAN must not overlap. That is, the IP address of a particular host cannot appear in two IP subnets.

The **vlan** parameter specifies the name or numerical VLAN Identifier of the IP subnet-based or protocol-based VLAN. The name is not case sensitive, however, the case is preserved for display purposes. The VLAN specified must exist.

The **subnet** parameter specifies the IP subnet address for the subnet.

The **mask** parameter specifies the network mask for the subnet address specified in the **subnet** parameter. The value must be consistent with the value specified for the **subnet** parameter. The default is the network mask for the address class of the IP address, (for example, 255.255.0.0 for a Class B address, 255.255.255.0 for a Class C address).

Use the **add vlan port** command to associate the specified IP subnet with one or more ports in the VLAN.

**Examples** To add the IP subnet 202.36.10 with mask 255.255.255.0 to the Marketing IP subnet-based VLAN, use the command:

```
add vlan=marketing subnet=202.36.1.0 mask=255.255.255.0
```

**Related Commands**

- [add vlan port](#)
- [delete vlan port](#)
- [delete vlan subnet](#)
- [show vlan](#)



## create switch trunk

**Syntax for x900-24X** `CREate SWitch TRunk=trunk [Port=port-list]  
[Speed={10M|100M|1000M|10G}]  
[THRASHAction={LEarndisable|LINKDown|NONE|Portdisable|  
VLANdisable}] [THRASHTimeout={None|1..86400}]`

**Syntax for x900-48FE and AT-9900** `CREate SWitch TRunk=trunk [Port=port-list]  
[Speed={10M|100M|1000M}] [THRASHAction={LEarndisable|  
LINKDown|NONE|Portdisable|VLANdisable}]  
[THRASHTimeout={None|1..86400}]`

where:

- *trunk* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen.
- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.

**Description** This command creates a trunk group on the switch and optionally adds ports to the trunk group and sets port speed. Port trunking **must** be configured on both ends of a link or network loops may result. The switch supports static 802.3ad link aggregation, and port trunking is also called *link aggregation* (for details, see [“Link Aggregation” on page 7-12](#)).

If you specify a trunk without **port** and **speed** parameters, a trunk group is created with an undefined speed.

The **trunk** parameter specifies a unique name for the trunk group. The name is not case sensitive although the case you enter is preserved for display purposes.

The **port** parameter specifies the switch ports to be added to the trunk group. The ports specified:

- must not be in another trunk group
- must have the same VLAN configuration
- must have identical STP port configurations
- cannot include the switch’s mirroring port
- are subject to the above constraints, and can have the following:

| Switch    | Max No. of Trunks | Max No. of Ports per Trunk |
|-----------|-------------------|----------------------------|
| AT-8948   | 7                 | 4                          |
| x900-48   | 7                 | 8                          |
| AT-9900   | 12                | 4                          |
| AT-9924Ts | 12                | 4                          |
| x900-24X  | 12                | 8                          |

When implementing private VLANs, note that it is not possible to have more than one uplink port in a VLAN unless the uplink ports are members of the same trunk group. See [“Private VLANs and port trunking” on page 7-32](#) for more information.

When you add a port to a trunk group, the switch saves the port's current speed and duplex mode settings and sets the port to autonegotiate to the speed of the trunk group and full duplex mode. If the port does not support auto-negotiation, the switch sets the port to the fixed speed of the trunk group and full duplex mode.

The **speed** parameter lets you specify the speed of ports in a trunk group. Note the following:

- The **10M** and **100M** options are valid for 10/100 RJ-45 ports on x900-48FE switches.
- The **10M**, **100M**, and **1000M** options are valid for 10/100/1000 RJ-45 ports on x900-24X and AT-9900 switches.
- The **10M**, **100M**, and **1000M** options are valid for 10/100/1000 copper SFP ports on x900-24X and AT-9900 switches.
- The **100M** option is valid for 100Mb fibre SFP ports on AT-9900 switches.
- The **1000M** option is valid for copper SFP ports on x900-48FE switches.
- The **100M** option is valid for 100Mb SFP ports on x900-24XS switches.
- The **1000M** option is valid for 1Gb fibre SFP ports on all models.
- The **10G** option is valid for 10Gb fibre XFP ports only on x900-24X switches.

If the **speed** parameter is not set, the trunk speed becomes the default speed of the master trunk port. The master trunk port is the lowest numbered port in the trunk group. The default is the highest speed the port is capable of.

The **thrashaction** parameter specifies the action the switch takes when it detects MAC address thrashing on a trunk. Thrashing occurs when one or more ports or trunks repeatedly learn the same MAC addresses, for example, as a result of a network loop. The switch applies the trunk's **thrashaction** to all ports in the trunk.

Take care with the **thrashaction** parameter because misuse can impair your network operation.

Set the **thrashaction** parameter to:

- **none** to apply no thrash limiting on the trunk.
- **learndisable** to disable MAC address learning on all ports in the thrashing trunk, until the period specified with the **thrashtimeout** parameter has elapsed. The default is **learndisable**.
- **portdisable** or **linkdown** to disable all ports in the thrashing trunk until either the period specified by the **thrashtimeout** parameter has elapsed, or until the ports or subset of ports in the trunk are re-enabled by the **enable switch port** command. If **linkdown** is specified, the link state is down; if **portdisable** is specified, the link state remains up.
- **vlandisable** to block all traffic on the VLAN where the address was learned, on all ports in the thrashing trunk, until either the period specified by **thrashtimeout** has elapsed, or until the ports are re-enabled using the **enable switch port vlan** command. When **thrashaction=vlandisable**, there is only one timer per trunk, so if multiple VLANs have been disabled on a trunk, the timer starts when the last VLAN was disabled. When the timer expires, all VLANs are re-enabled on the trunk. When **thrashaction=vlandisable**, ingress filtering is automatically enabled on all ports in the trunk.

The **thrashtimeout** parameter specifies the time, in seconds, for which the switch employs the thrash action specified by the **thrashaction** parameter. The **thrashtimeout** cannot be set to **none** if **thrashaction=learndisable**. If **thrashtimeout=none**, and **thrashaction** is then changed to **learndisable**, then the switch automatically changes the **thrashtimeout** to 1 second.

If **none** is specified, the trunk is not automatically re-enabled, but individual ports can be re-enabled by using the **enable switch port** command for **thrashaction=portdisable** or **linkdisable**, and the **enable switch port vlan** command for **thrashaction=vlandisable**. The default is 1 second.

**Example** To create a trunk group called Trunk1 containing ports 1 to 4, use the command:

```
cre swi tr=Trunk1 po=1-4
```

**Related Commands**

- [add switch trunk](#)
- [delete switch trunk](#)
- [destroy switch trunk](#)
- [enable switch port](#)
- [enable switch port vlan](#)
- [set switch trunk](#)
- [show switch trunk](#)

## create vlan

**Syntax** CREate VLAN=*vlan-name* VID=2..4094 [PRIVate]

CREate VLAN=*vlan-name* VID=2..4094 SUBNET=*ipadd*  
[MASK=*ipadd*] [PRIVate]

CREate VLAN=*vlan-name* VID=2..4094 PROTOcol=*protocol-type*  
[PRIVATE]

CREate VLAN=*vlan-name* VID=2..4094 [NESTed]

CREate VLAN=*vlan-name* VID=2 4094 SUBNET=*ipadd* [MASK=*ipadd*]  
[NESTed]

CREate VLAN=*vlan-name* VID=2..4094 PROTOcol=*protocol-type*  
[NESTed]

where:

- *ipadd* is an IP address in dotted decimal notation
- *protocol-type* is either a valid protocol number, or a recognised protocol name. A protocol number can be either 1 byte for SAP, 2 bytes for ETHII or 5 bytes for an 802.3/802.2 SNAP type packet, and is specified in hexadecimal. Hexadecimal numbers must begin with "0x".
- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all** or **default**.

**Description** This command creates a static VLAN with a unique name and VLAN Identifier (VID), and assigns it to the default STP. Different types of VLAN can be created, subject to the subset of VLANs available at any one time. The following table describes the types of VLAN available with each VLAN classification.

| This type of VLAN is available... | For this VLAN classification... |
|-----------------------------------|---------------------------------|
| IP Subnet<br>Protocol<br>Port     | IP subnet-based                 |
| Protocol<br>Port                  | Protocol-based                  |
| Port                              | Port-based                      |

To change the VID of an existing VLAN, the VLAN must be destroyed and created again with the new VID.

If you create a private VLAN, the ports you add to it are unable to communicate with each other, but can access another network. See [“Private VLANs” on page 7-30](#) for information about configuring private VLANs.

The **show switch** command displays which VLAN classification rules are being used by the switch.

When the switch is powered up, and before any configuration script commands are executed, a default VLAN, whose type has yet to be defined, is created. All ports in the switch are added to the default VLAN, using the Port VLAN classification rule.

To use the IP subnet and Protocol VLAN classification rules, add an IP subnet-based or protocol-based VLAN as the first type of non port-based VLAN created.

If the VLAN created is not port-based, then the user may optionally add one of the following:

- an IP subnet-based VLAN by specifying the **subnet** and **mask** parameters
- a Protocol-based VLAN by specifying the **protocol** parameter

The **vlan** parameter specifies a unique name for the VLAN. This name can be more meaningful than the VID, to make administration easier. The VLAN name is only used within the switch; it is not transmitted to other VLAN-aware devices, or used in the forwarding process or stored in the forwarding database. If the VLAN name begins with “vlan” and ends with a number, such as “vlan1” or “vlan234”, then the number must be the same as the **vid** specified. This avoids confusion when identifying to which VLAN subsequent commands refer.

The **vid** parameter specifies a unique VLAN Identifier for the VLAN. If VLAN tagged ports are added to this VLAN, the specified VID is used in the VID field of the tag in outgoing frames. If untagged ports are added to this VLAN, the specified VID only acts as an identifier for the VLAN in the forwarding database. The default port based VLAN has a VID of 1.

The **subnet** and **mask** parameters add an IP subnet to the VLAN after the VLAN is created. See the [add vlan subnet command on page 7-64](#) for more details.

The **protocol** parameter adds a protocol to the VLAN after the VLAN is created. See the [add vlan protocol command on page 7-62](#) for more details.

The **private** parameter specifies that the VLAN is a private VLAN. In a private VLAN, all traffic received on a given port is sent to a pre-defined uplink port regardless of the MAC destination address or VLAN. See “Private VLANs” on [page 7-30](#) for more information.

The **nested** parameter specifies that the VLAN is a nested VLAN. Depending on whether traffic is ingressing or egressing a Core port a second VID, termed a Customer-ID (CID), is either inserted or removed. See “Nested VLANs” on [page 7-32](#) for more information.

**Examples** To create a VLAN named Marketing with a VLAN identifier of 2, use the command:

```
cre vlan=marketing vid=2
```

To create a VLAN named vlan42, which must have a VID of 42, use the command:

```
cre vlan=vlan42 vid=42
```

To create an IP subnet-based VLAN named Fred, with a VID of 123 and an IP subnet address 202.36.1.0 (and default mask), use the command:

```
cre vlan=fred vid=123 subnet=202.36.1.0
```

**Related Commands**

- [add vlan port](#)
- [add vlan protocol](#)
- [add vlan subnet](#)
- [show switch](#)
- [show vlan](#)

## delete lacp port

---

**Syntax** DELEte LACP PORt={*port-list*}

where *port-list* is a port number, range (specified as *n-m*), or comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered switch port, including uplink ports.

**Description** This command removes ports from LACP's control and LACP frames are no longer transmitted across the link. It is good practice to delete LACP from ports that are linked to non-LACP-capable devices.

The **port** parameter specifies switch ports to be deleted from LACP's control. Ports specified must be under the control of LACP. **all** is not a configurable option; to stop LACP on all ports, use the **disable lacp** command.

**Examples** To delete ports 3 and 5 from LACP, use the command:

```
del lacp po=3,5
```

**Related Commands**

- [add lacp port](#)
- [disable lacp](#)
- [disable stp](#)
- [set lacp port](#)
- [show lacp port](#)

## delete switch accelerator hwfilter

---

**Syntax** DELEte SWItch ACCELerator HWFilter=1..2000  
DELEte SWItch ACCELerator HWFilter=ALL

**Description** This command deletes one or all hardware-based packet filters from the IPv6 accelerator card on AT-8948 switches.

The **hwfilter** parameter specifies the numerical identifier of an existing hardware-based packet filter.

**Example** To delete the hardware-based packet filter with identifier 2, use the command:

```
del swi accel hwf=2
```

**Related Commands**

- [add switch accelerator hwfilter](#)
- [show switch accelerator](#)
- [show switch accelerator counter](#)
- [show switch accelerator hwfilter](#)

---

## delete switch filter

---

**Syntax** `DELEte SWItch FILter PORT=port ENTRy=entry-list`

where:

- *entry-list* is an entry number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Entry numbers start at 0 and end at *m*, where *m* is the highest filter entry currently defined in the permanent forwarding database. Each port has its own permanent forwarding database.
- *port* is the number of one of the switch ports or an uplink port.

**Description** This command deletes specific static filter entry ports from the permanent forwarding database. The static filter is deleted on the port specified by the **port** parameter. The **entry** parameter must specify an existing filter entry in the permanent forwarding database.

**Example** To delete filter entry 9 on port 2, use the command:

```
del swi fil po=2 ent=9
```

**Related Commands** [add switch filter action](#)  
[show switch filter](#)

---

## delete switch hwfilter

---

**Syntax** `DELEte SWItch HWFilter=filter-idlist`

where *filter-idlist* is a filter ID, range (specified as *n-m*), or comma-separated list of filter IDs and/or ranges. FilterIDs start at 1.

**Description** This command deletes one or more hardware-based packet filters from the switch.

The **hwfilter** parameter specifies the numerical identifier of one or more existing hardware-based packet filters.

**Examples** To delete the hardware-based packet filters with identifiers 2 and 5, use the command:

```
del swi hwf=2,5
```

**Related Commands** [add switch hwfilter](#)  
[show switch hwfilter](#)

## delete switch trunk

---

**Syntax** DELEte SWItch TRunk=*trunk* PORT={*port-list*|ALL}

where:

- *trunk* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen.
- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered switch Ethernet port, including uplink ports.

**Description** This command deletes ports from an existing trunk group on the switch.

The **trunk** parameter specifies the name of the trunk group. The name is not case sensitive. The name uniquely identifies the trunk group. The specified trunk group must already exist.

The **port** parameter specifies switch ports to be deleted from the trunk group. Ports specified must be in the specified trunk group. If **all** is specified, then all ports in the trunk group are deleted.

A port in a trunk group cannot be deleted if the trunk group is the uplink ports of a private VLAN. See [“Private VLANs and port trunking” on page 7-32](#) for more information.

When you remove a port from a trunk group, the switch restores the port's speed and duplex mode settings to the values they had before the port was added to the trunk group.

**Example** To delete port 3 from trunk group Trunk1, use the command:

```
del swi tr=Trunk1 po=3
```

**Related Commands**

- [add switch trunk](#)
- [create switch trunk](#)
- [destroy switch trunk](#)
- [set switch trunk](#)
- [show switch trunk](#)



## delete vlan port

**Syntax** DELEte VLAN={*vlan-name*|1..4094} PORt={*port-list*|ALL}

DELEte VLAN={*vlan-name*|1..4094} PORt={*port-list*|ALL}  
SUBNET={*ipadd*|ALL}

DELEte VLAN={*vlan-name*|1..4094} PORt={*port-list*|ALL}  
PROTOcol={*protocol-type*|*index-list*|ALL}

where:

- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.
- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port.
- *ipadd* is an IP address in dotted decimal notation.
- *protocol-type* is either a valid protocol number or a recognised protocol name. A protocol number can be 1 byte for SAP, 2 bytes for ETHII, or 5 bytes for an 802.3/802.2 SNAP packet. The protocol-type is specified as a hexadecimal number, which begins with "0x".
- *index-list* is a single index number or group as either a comma-separated list, range (specified as *n-m*), or a combination of the two. Index numbers start at 0.
- *group-number* is a single numerical number.
- *uplink-number* is a string beginning with the name UPLINK and concatenated at the end with a numerical identifier. For example UPLINK1.

**Description** This command deletes ports from the specified VLAN. An untagged port can be deleted from a VLAN if the port is still a member of a VLAN after the deletion has occurred. If the deleted port does not belong to any static VLAN as a tagged port and the port is disassociated from the Port VLAN classification rule of the VLAN, then the port is implicitly added to the default VLAN as an untagged port and associated with the Port VLAN classification rule of the default VLAN. It is not possible to delete a port that belongs only to the default VLAN.

If the port becomes only a tagged port as a result of the deletion, that is, the port does not belong to any VLAN as an untagged port, then the **acceptable** switch parameter for the port is set to VLAN. The user cannot change the **acceptable** parameter for the port.

A tagged port can be deleted from a VLAN if the port is still a member of a VLAN after the deletion has occurred.

If as a result of the port deletion, ports are moved from one STP to another STP, the two affected STPs are initialised if they are presently enabled. Any previously disabled ports in the STPs are enabled.

This command can also be used to disassociate a port that transmits untagged packets for the VLAN from an IP subnet or protocol belonging to the VLAN, or from the Port VLAN classification rule of the VLAN. If as a result of the disassociation, the port is not associated with any VLAN classification rule of the VLAN, then the port is deleted from the VLAN.

The **vlan** parameter specifies the name or numerical VLAN Identifier of the VLAN. The name is not case sensitive. The VLAN must already exist. If the **vlan** parameter specifies **default** or **1**, that is the default VLAN, and the type of the default VLAN changes to undefined following this command, that is only port-based VLANs exist in the switch and the default VLAN uses only instances of the Port VLAN classification rule, then the switch is in a state where the types of VLAN that can be created can be implicitly set.

The **port** parameter specifies the ports to be deleted from the VLAN. If **all** is specified, then all ports belonging to the VLAN are deleted. If the command would succeed on a subset of the ports specified, but cause an error on the others, then the command fails and has no effect.

If a port belongs to a trunk group, all the ports in the trunk group must be specified. A subset of the ports in a trunk group cannot be deleted from the VLAN unless they are first removed from the trunk group.

A private VLAN cannot contain any private ports when an uplink port or trunk group of ports is deleted from the VLAN because a private VLAN must always contain one uplink port, or trunk group of ports. To delete the uplink port or trunk group of ports from a private VLAN, first delete any private ports from the VLAN.

The **subnet** parameter is used to disassociate an IP subnet from a port if the port transmits untagged packets for the VLAN. If the port is not associated with any VLAN classification rule of the VLAN after the disassociation, the port is deleted from the VLAN. If the **subnet** parameter specifies **all**, all previously added IP subnets are disassociated from the port.

The **protocol** parameter is used to disassociate a protocol from a port if the port transmits untagged packets for the VLAN. If the port is not associated with any VLAN classification rule of the VLAN after the disassociation, then the port is deleted from the VLAN. If the **protocol** parameter specifies an index number, the index number represents a protocol previously added to the VLAN. If the **protocol** parameter specifies **all**, all previously added protocols are disassociated from the port.

Combinations of the optional **subnet** or **protocol** parameters cannot be specified on the command line at the same time.

A port that transmits untagged packets for a VLAN must be associated with at least one IP subnet or protocol belonging to the VLAN, or be associated with the Port VLAN classification rule of the VLAN. The association between a port and a particular IP subnet or protocol can only occur for one VLAN to which the port belongs.

**Example** To delete ports 3 and 6 from the Accounting VLAN, use the command:

```
del vlan=accounting po=3,6
```

To delete all ports from the Sales VLAN, use the command:

```
del vlan=sales po=all
```

**Related Commands**

- [delete vlan protocol](#)
- [delete vlan subnet](#)
- [set vlan port](#)
- [show vlan](#)

---

## delete vlan protocol

---

**Syntax** DELEte VLAN={*vlan-name*|1..4094} PROTOcol={*protocol-type*|*index-list*|ALL}

where:

- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.
- *protocol-type* is either a valid protocol number or a recognised protocol name. A protocol number can be either 1 byte for SAP, 2 bytes for ETHII, or 5 bytes for an 802.3/802.2 SNAP type packet, and is specified in hexadecimal. Hexadecimal numbers must begin with "0x"
- *index-list* is a single index number or group as either a comma-separated list, range (specified as *n-m*), or a combination of the two. Index numbers start at 0.

**Description** This command deletes one or more protocol rules from a VLAN. If the protocol has been associated with a port in the VLAN, then the command fails. After deletion of the protocol(s), the index numbers for the remaining protocols are updated. The index numbers form a contiguous range.

The **vlan** parameter specifies the name of the VLAN, or the numerical VLAN Identifier (VID) of the VLAN. The name is not case sensitive; however, the case is preserved for display purposes. The VLAN specified must exist.

The **protocol** parameter specifies an Ethernet protocol. The VLAN module provides a pre-defined list of common protocols. Alternatively, the **protocol** parameter can specify the index number of a protocol that was added to the VLAN. If **all** is specified, then all protocols that have been added to the VLAN are deleted.

Use the **delete vlan port** command to delete an association between a protocol and port in the VLAN without deleting the protocol from the VLAN.

**Examples** To delete the IP protocol (index number 0) from the Marketing protocol-based VLAN, use one of the following commands:

```
del vlan=marketing prot=ip
del vlan=marketing prot=0X0800
del vlan=marketing prot=0
```

To delete the IPX protocol for Ethernet-II packets (index number 1) from the Sales IP subnet-based VLAN, use one of the following commands:

```
del vlan=sales prot=ipx
del vlan=sales prot=0X8137
del vlan=sales prot=1
```

**Related Commands**

- [add vlan port](#)
- [add vlan protocol](#)
- [delete vlan port](#)
- [show vlan](#)

## delete vlan subnet

---

**Syntax** DELEte VLAN={*vlan-name*|1..4094} SUBNET={*ipadd*|ALL}

where:

- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.
- *ipadd* is an IP address in dotted decimal notation.

**Description** This command deletes one or all IP subnets from a VLAN. If the IP subnet has been associated with a port in the VLAN, then the command fails.

The **vlan** parameter specifies the name of the IP subnet-based VLAN or the numerical VLAN identifier of the IP subnet-based VLAN. The name is not case sensitive; however, the case is preserved for display purposes. The VLAN specified must exist.

The **subnet** parameter specifies the IP subnet address for the subnet. If **all** is specified, then all IP subnets previously added to the VLAN are deleted.

Use the **delete vlan port** command to delete an association between a protocol and port in the VLAN without deleting the protocol from the VLAN.

**Examples** To delete the IP subnet 202.36.1.0 with mask 255.255.255.0 (index number 2) from the Marketing IP subnet-based VLAN, use the following command:

```
del vlan=marketing subnet=202.36.1.0
```

To delete all IP subnets from the *sales* IP subnet-based VLAN, use the following command:

```
del vlan=sales subnet=all
```

**Related Commands**

- [add vlan port](#)
- [add vlan subnet](#)
- [delete vlan port](#)
- [show vlan](#)

---

## destroy switch trunk

---

**Syntax** DESTroy SWItch TRunk=*trunk*

where *trunk* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen.

**Description** This command destroys a trunk group on the switch. The trunk group must be empty; that is, it must not contain any ports.

The **trunk** parameter specifies the name of the trunk group. The name is not case sensitive. The name uniquely identifies the trunk group. The specified trunk group must already exist.

**Example** To destroy a trunk group called Trunk1, use the command:

```
dest swi tr=Trunk1
```

**Related Commands** [add switch trunk](#)  
[create switch trunk](#)  
[delete switch trunk](#)  
[set switch trunk](#)  
[show switch trunk](#)

---

## destroy vlan

---

**Syntax** DESTroy VLAN={*vlan-name* | 2..4094 | ALL}

where *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all** or **default**.

**Description** This command destroys the specified static VLAN or all static VLANs in the switch.

The **vlan** parameter specifies the name of the VLAN or the numerical VLAN Identifier (VID) of the VLAN. The name is not case sensitive although the case is preserved for display purposes. The **vlan** specified must exist.

The default VLAN, which has a numerical VLAN Identifier (VID) of 1, cannot be destroyed. If **all** is specified for this parameter, all static VLANs are destroyed except for the default VLAN. A VLAN cannot be destroyed when ports still belong to it, or if other modules, except GARP, are attached to it.

**Examples** To destroy the VLAN with the VLAN Identifier of 1234, use the command:

```
dest vlan=1234
```

To remove all user created VLANs from the switch, none of which have any member ports, use the command:

```
dest vlan=all
```

**Related Commands** [create vlan](#)  
[show vlan](#)

## disable lacp

---

**Syntax**    DISable LACP

**Description**    This command disables LACP processes on the switch. A warning, notification, and log message are generated when this command is executed. LACP is disabled by default. LACP port settings that are changed while LACP is disabled take effect when LACP is re-enabled.

**Related Commands**    [enable lacp](#)  
[show lacp](#)

## disable lacp debug

---

**Syntax**    DISable LACP DEBug={MSG|PACKet|STATe|TRAcE|DEV|PERSistent|ALL}

**Description**    This command disables the LACP debugging process, which is disabled by default. The **msg** option displays the decoded form of incoming and outgoing LACP packets. The **packet** option displays incoming and outgoing LACP packets in hex. The **state** option displays internal state machine changes. The **trace** option displays the function call tree. The **persistent** option enables the debug state to persist over one reboot. If all is specified, the debugging process is disabled for all options. The default is **all**.

**Related Commands**    [enable lacp debug](#)  
[show lacp](#)  
[disable debug active](#) in Chapter 4, Configuring and Monitoring the System  
[show debug active](#) in Chapter 4, Configuring and Monitoring the System

## disable switch accelerator

---

**Syntax**    DISable SWItch ACCELerator

**Description**    This command disables the IPv6 accelerator card on AT-8948 switches, which is enabled by default. The command takes effect after a reboot, and results in the software processing packets instead of the IPv6 accelerator card.

For more information about accelerated IPv6 traffic, see “[IPv6 Acceleration](#)” on page 23-21 of Chapter 23, Internet Protocol version 6 (IPv6).

**Examples**    To disable the IPv6 accelerator card, use the following command and then reboot the switch:

```
dis swi accel
```

**Related Commands**    [enable switch accelerator](#)  
[show switch accelerator](#)

---

## disable switch accelerator function

---

**Syntax** `DISable SWITch ACCELerator FUNction=ICMPredirect`

**Description** This command stops the switch from issuing ICMP redirect messages for unicast IPv6 traffic on AT-8948 switches. The **icmpredirect** function is disabled by default.

**Examples** To stop ICMP redirect messages from being issued, use the command:

```
dis swi accel func=icmp
```

**Related Commands** [enable switch accelerator function](#)  
[show switch accelerator](#)

---

## disable switch accelerator debug

---

**Syntax** `DISable SWITch ACCELerator DEBug={ACCElerator|ALL|BIST|  
INTERnal|INT_APi|INT_BLd|INT_CFg|INT_DEf|INT_ERr|  
INT_FNC|INT_HMm|INT_MUx|INT_PRm|INT_STR|INT_TRc|  
INT_USr|MALloc|None|PCI|PCIPrim|SMEM|STUB|USEful}  
[PERsistent]`

**Description** This command disables debugging output for the IPv6 accelerator card on AT-8948 switches. Debugging may or may not already be enabled. Selected debugging options are removed from the existing set of options. By default, all debugging is disabled.

For more information about accelerated IPv6 traffic, see [“IPv6 Acceleration” on page 23-21 of Chapter 23, Internet Protocol version 6 \(IPv6\)](#).

The **debug** parameter specifies which debugging options are to be disabled.

The **persistent** parameter turns off persistent debugging. When persistent debugging is enabled, the debug mode and timeout set is retained across reboots of the switch. This automatically enables debug in a switch restart with debug output being sent to the first asynchronous port.

**Examples** To disable the display of debugging for memory usage

```
dis swi accel deb=mal
```

**Related Commands** [enable switch accelerator debug](#)  
[show switch accelerator counter](#)  
[show switch accelerator debug](#)

## disable switch ageingtimer

---

**Syntax** DISable SWItch AGEingtimer

**Description** This command disables the ageing timer from ageing out dynamically learned entries in the forwarding database. The default setting for the ageing timer is enabled.

**Example** To disable the ageing out of learned MAC addresses, use the command:

```
dis swi age
```

**Related Commands** [enable switch ageingtimer](#)  
[set switch ageingtimer](#)  
[show switch](#)

## disable switch debug

---

**Syntax** DISable SWItch DEBUg={DMA | QOS | PHY | ALL}

**Description** This command disables a specific debug mode or all debugging on the switch. The **debug** parameter specifies which mode to disable. The following table describes available options.

| Debug Options | Description                                          |
|---------------|------------------------------------------------------|
| DMA           | Operations related to Direct Memory Access requests. |
| QOS           | Operations related to Quality of Service             |
| PHY           | Operations related to the PHY port interfaces.       |
| ALL           | All debug options                                    |

**Example** To disable all switch debugging, use the command:

```
dis swi deb=all
```

**Related Commands** [enable switch debug](#)  
[show switch](#)  
[disable debug active in Chapter 4, Configuring and Monitoring the System](#)  
[show debug active in Chapter 4, Configuring and Monitoring the System](#)



## disable switch hash

**Syntax** DISable SWItch HASH={L2|L3|L4}[,...]

**Description** This command disables the use of packet fields in the Link Aggregate hash function. All packet fields can be disabled.

The **hash** parameter specifies the type of packet fields to use in the Link Aggregate hash function, as a comma-separated list. The default is to stop using all fields. The following table describes individual fields.

| Value | Packet Field                                                                                                                       |
|-------|------------------------------------------------------------------------------------------------------------------------------------|
| L2    | Source and destination MAC addresses.                                                                                              |
| L3    | Source and destination IP addresses<br>Destination IPX network addresses                                                           |
| L4    | Source and destination TCP or UDP ports for TCP/IP and UDP/IP packets<br>Source and destination IPX socket numbers for IPX packets |

**Examples** To disable the use of Layer 4 fields in a packet in the Link Aggregate hash function, use the command:

```
dis swi hash=L4
```

To disable the use of Layer 2 and Layer 3 fields in a packet in the Link Aggregate hash function, use the command:

```
dis swi hash=L2,L3
```

**Related Commands** [enable switch hash](#)  
[show switch](#)

## disable switch mclimiting

**Syntax** DISable SWItch MClimiting

**Description** This command disables multicast packet limiting for all ports on the switch. Ports with broadcast and multicast rate limiting enabled change so that they have broadcast packet limiting enabled.

**Examples** To disable the limiting of reception of multicast packets, use the command:

```
dis swi mc
```

**Related Commands** [enable switch mclimiting](#)  
[set switch port](#)

## disable switch learning

---

**Syntax**    DISable SWItch LEarning

**Description**    This command disables the dynamic learning and updating of the forwarding database. The default setting for the learning function is enabled.

If switch learning is disabled and the ageing timer has aged out all dynamically learned filter entries, only statically entered MAC source addresses decide which packets to forward or discard. When the switch finds no matching entries in the forwarding database during the forwarding process, all switch ports in the VLAN are flooded with the packet, except the port that received it.

**Example**    To disable the switch learning function, use the command:

```
dis swi le
```

**Related Commands**    [enable switch learning](#)  
[show switch](#)

## disable switch mirror

---

**Syntax**    DISable SWItch MIRRor

**Description**    This command disables traffic mirroring on the switch. Mirrored traffic stops being sent on the mirror port. The mirror port and mirror settings for the sources of mirror traffic remain configured. The default state of switch mirroring is disabled.

**Example**    To disable traffic mirroring, use the command:

```
dis swi mirr
```

**Related Commands**    [enable switch mirror](#)  
[set switch mirror](#)  
[set switch port](#)  
[show switch](#)  
[show switch port](#)

## disable switch port

---

**Syntax** DISable SWItch Port={*port-list*|ALL} [AUTOmDi]  
[EGReSSqueue=*queue-list*] [FLOw=PAUSE|JAMMING]  
[LiNK={DISable|ENABle}]

where

- *port-list* is a single port number or a group as either a comma-separated list, range (specified as *n-m*), or a combination of the two. Port numbers start at 1 and end at *m*, where *m* is the highest numbered switch Ethernet port.
- *queue-list* is an egress queue number, a range of queue numbers (specified as *n-m*), or a comma separated list of queue numbers and/or ranges. Egress queue numbers start at 0 and end at 7.

**Description** This command disables one or more of the following:

- a port or group of ports on the switch (switch ports are enabled by default)
- a port or group of ports, including its link. This ensures that the device at the other end of the link realises that the port is down.
- auto MDI/MDI-X mode on the ports
- flow control on the ports
- egress queues on the ports

When you disable a port it no longer sends or receives frames. When you disable egress queues, frames scheduled for those queues are no longer transmitted. Note that disabling a port does not disable STP operation on the port. Ports should be disabled if there is faulty wiring or equipment attached to the ports, or as a security measure to stop access from intruders. Switch ports are enabled by default.

The **port** parameter specifies the port to disable.

The **automdi** parameter disables auto MDI/MDI-X mode, and sets the polarity to the default of MDI-X. Auto MDI/MDI-X mode is enabled by default.

The **egressqueue** parameter specifies the egress queues to disable. If you specify this parameter, you must supply a value for it. At switch startup all egress queues are enabled by default.

The **flow** parameter specifies the type of flow control to be disabled for the port. If **pause** is specified, pause frames are disabled, which control flow for full duplex ports. If **jamming** is enabled, jamming signals are disabled, which control flow for half duplex ports. Both forms of flow control are disabled by default.

The **link** parameter specifies whether the ports are enabled or disabled at the hardware level. If **disable** is specified, this is the same as disconnecting the cable or turning off the port optics. If the **link** parameter is not specified, the link remains physically enabled. On a disabled port, entering the command **disable switch port=port-number link=enable** brings the link up in software without re-enabling the port.

**Examples** To disable ports 2, 4 and 6 at the hardware level, use the command:

```
dis swi po=2,4,6 lin=dis
```

To disable auto MDI/MDI-X on ports 2 and 4, use the command:

```
dis swi po=2,4 autom
```

To disable egress queues 0, and 3 to 5 on port 1, use the command:

```
dis swi po=1 egr=0,3-5
```

To disable pause frames on port 1, use the command:

```
dis swi po=1 flo=pause
```

**Related Commands** [enable switch port](#)  
[set switch port](#)  
[show switch port](#)

## disable switch port vlan

**Syntax** DISable SWItch PORt={*port-list* | ALL} VLAN[={*vlan-name* | 1..4094 | ALL}]

where:

- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.
- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen.

**Description** This command disables one or more ports from VLANs to which they belong. Once disabled, a port remains a member of the VLAN, but does not receive or transmit packets from that VLAN.

The **port** parameter specifies the port or ports to disable. If a trunked port is specified, all ports in the trunk are disabled. When a VLAN is disabled on a port, ingress filtering is automatically enabled for that port

The **vlan** parameter specifies the VLAN or VLANs for which ports are disabled. Specified ports must be a member of the VLAN. If no value, or **all** is specified, the specified ports will be disabled for all VLANs to which they belong.

**Examples** To disable the default vlan on port 1, use the command:

```
dis swi po=1 vlan=1
```

**Related Commands** [enable switch port vlan](#)  
[show switch port](#)

---

## disable vlan debug

---

**Syntax** DISable VLAN={*vlan-name* | 1..4094 | ALL} DEBug={PKT | ALL}

where *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.

**Description** This command disables packet debugging or all debugging for the specified VLAN or all VLANs. The default is for all VLAN debugging to be disabled.

The **debug** parameter specifies the VLAN debugging mode to be disabled. If **pkt** is specified, the packet debug mode (displaying raw ASCII packets) is disabled. If **all** is specified, all debugging is disabled.

**Example** To disable packet debugging on the Marketing VLAN, use the command:

```
dis vlan=marketing deb=pkt
```

**Related Commands** [enable vlan debug](#)  
[show vlan port](#)

---

## enable lacp

---

**Syntax** ENABle LACP

**Description** This command enables LACP on the switch. A notification message and a log message file are generated when this command is executed. LACP is disabled by default.

**Related Commands** [disable lacp](#)  
[show lacp](#)

---

## enable lacp debug

---

**Syntax** ENABle LACP DEBug={MSG | PACKet | STATe | TRAcE | DEV | PERSistent | ALL}

**Description** This command enables the LACP debugging facility, which is disabled by default. The **msg** option displays the decoded form of incoming and outgoing LACP packets. The **packet** option displays all incoming and outgoing LACP packets. The **state** option displays internal state machine changes. The **trace** option displays the function call tree. The **dev** option displays internal support information. The **persistent** option enables the debug state to persist over one reboot. If **all** is specified, the debugging process is enabled for all options. The default is **all**.

**Related Commands** [disable lacp debug](#)  
[show lacp](#)  
[disable debug active](#) in Chapter 4, Configuring and Monitoring the System  
[show debug active](#) in Chapter 4, Configuring and Monitoring the System

## enable switch accelerator

---

**Syntax** ENABle SWITch ACCELeRator

**Description** This command enables the IPv6 accelerator card on AT-8948 switches so that the card handles IPv6 traffic routing functions. This command takes effect after a reboot. The IPv6 accelerator card is enabled by default.

For more information about accelerated IPv6 traffic, see [“IPv6 Acceleration” on page 23-21 of Chapter 23, Internet Protocol version 6 \(IPv6\)](#).

**Examples** To enable the IPv6 accelerator card, use the command:

```
ena swi accel
```

and then reboot the switch.

**Related Commands** [disable switch accelerator](#)  
[show switch accelerator](#)

## enable switch accelerator function

---

**Syntax** ENABle SWITch ACCELeRator FUNCtion=ICMPredirect

**Description** This command enables the AT-8948 switch to issue ICMP redirect messages for unicast IPv6 traffic as recommended by RFC 2461, *Neighbor Discovery for IP Version 6 (IPv6)*. The **icmpredirect** function is disabled by default.

The switch accelerator does not issue ICMP redirect messages. Instead, when **icmpredirect** is enabled, any unicast IPv6 packet that has the same ingress and egress VLAN is routed in software rather than hardware, and the software may generate ICMP redirect messages.

The purpose of an ICMP redirect is to replace layer 3 routing with Layer 2 switching. On a layer 3 switch, switching and routing are both performed in hardware, so ICMP redirects offer no performance improvement. The **icmpredirect** function may impact performance in particular (unusual) network configurations, and we do not recommend it for use in a general network environment.

**Examples** To enable issuing of ICMP redirect messages, use the command:

```
ena swi accel func=icmp
```

**Related Commands** [disable switch accelerator function](#)  
[enable switch accelerator](#)  
[show switch accelerator](#)

## enable switch accelerator debug

**Syntax** `ENABle SWiTh ACCELerator DEBUg={ACCElerator|ALL|INTERNAL|INT_APi|INT_BLd|INT_CFG|INT_DEf|INT_ERr|INT_FNC|INT_HMm|INT_MUx|INT_PRm|INT_STR|INT_TRc|INT_USr|MALloc|NOne|PCI|PCIPrim|SMEM|STUB|USEful} [PERSistent] [TIMEOut={0..4000000000|NONne}]`

**Description** This command enables debugging output for the IPv6 accelerator card on AT-8948 switches. If debugging is already enabled, the command adds additional debugging options to the options set. Debugging is disabled by default. Debugging information is sent to the port or telnet session where the command originated.

For more information about accelerated IPv6 traffic, see [“IPv6 Acceleration” on page 23-21 of Chapter 23, Internet Protocol version 6 \(IPv6\)](#).

The **debug** parameter specifies the debugging options to enable. The **useful** option is a set of debug flags chosen to balance output size with useful detail.

The **persistent** parameter specifies that the debug mode and timeout set must be retained across reboots of the switch. This has the effect of automatically enabling debug in a switch restart, with debug output being sent to the first asynchronous port.

The **timeout** parameter specifies a time in seconds after which debugging automatically ceases. If **0** or **none** is specified, debugging must be disabled manually. The switch retains the value that you specify for **timeout** the first time you enable an applicable debugging mode for future **enable switch accelerator debug** commands. The default is **none**.

This debugging is intended for internal use only. Enabling switch accelerator debug options can generate enormous amounts of output, causing the switch to lock up. We recommend that you limit the time with the **timeout** option.

**Examples** To enable the display of debugging for memory usage

```
ena swi accel deb=mal
```

**Related Commands** [disable switch accelerator debug](#)  
[show switch accelerator counter](#)  
[show switch accelerator debug](#)

## enable switch ageingtimer

---

**Syntax** ENABle SWITch AGEingtimer

**Description** This command enables the ageing timer to age out dynamically learned entries in the forwarding database. The default setting for the ageing timer is enabled.

If the ageing timer ages out all dynamically learned filter entries, and switch learning is disabled, only statically entered MAC source addresses decide the packets to forward or discard. When the switch finds no matching entries in the forwarding database during the forwarding process, all switch ports in the VLAN are flooded with the packet, except the port that received it.

**Example** To enable the ageing out of learned MAC addresses, use the command:

```
ena swi age
```

**Related Commands** [disable switch ageingtimer](#)  
[set switch ageingtimer](#)  
[show switch](#)

## enable switch bist

---

**Syntax** ENABle SWITch BIST INSTance={0..3} [FULl]

**Description** This command runs a set of built-in self tests on the switch, packet buffer SDRAM, and route SRAM, and produces output.

The **instance** parameter indexes a specific switch instance to test.

Disconnect the switch from the network before issuing this command, and restart the switch afterwards. Network and switch performance are affected by this command so only authorised service personnel should use it.

Figure 7-10: Example output from the **enable switch bist instance=0** command for x900-48FE and AT-9900 switches

```
Running BIST (Built In Self Test) on switch instance 0

INFO - This will take less than 1 minute

Step 1: Testing switch device register access ..... Passed
Step 2: Testing switch device external buffer memory .... Passed
Step 3: Testing switch device external route memory ..... Passed

INFO - Switch device 0 passed all tests successfully

WARNING - The switch must be restarted after running the BIST
```



Figure 7-11: Example output from the **enable switch bist instance=0** command for x900-24X switches

```
Running BIST (Built In Self Test) on switch instance 0

INFO - This will take less than 1 minute

Step 1: Testing switch device register access ..... Passed
Step 2: Testing switch device external buffer memory .... Passed
Step 3: Testing switch device external route memory ..... Passed
Step 4: Testing Fabric Adapter register access ..... Passed
Step 5: Testing Fabric Adapter external buffer memory ... Passed

INFO - Switch device 0 passed all tests successfully

WARNING - The switch must be restarted after running the BIST
```

Figure 7-12:

Figure 7-13: Example output from the **enable switch bist instance=0 full** command for x900-48FE and AT-9900 switches

```
Running full BIST (Built In Self Test) on switch instance 0

INFO - This will take approximately 4 minutes

Step 1: Testing switch device register access ..... Passed
Step 2: Testing switch device external buffer memory .... Passed
Step 3: Testing switch device external route memory ..... Passed

INFO - Switch device 0 passed all tests successfully

WARNING - The switch must be restarted after running the BIST
```

Figure 7-14: Example output from the **enable switch bist instance=0 full** command for x900-24X switches

```
Running full BIST (Built In Self Test) on switch instance 0

INFO - This will take approximately 12 minutes

Step 1: Testing switch device register access ..... Passed
Step 2: Testing switch device external buffer memory .... Passed
Step 3: Testing switch device external route memory ..... Passed
Step 4: Testing Fabric Adapter register access ..... Passed
Step 5: Testing Fabric Adapter external buffer memory ... Passed

INFO - Switch device 0 passed all tests successfully

WARNING - The switch must be restarted after running the BIST
```

## enable switch debug

**Syntax** ENABle SWitch DEBUg={DMA|QOS|PHY|ALL} [OUTput=CONSOLE]  
[TIMEOut={1..4000000000|NONE}]

**Description** This command enables a specific debug mode or all debugging on the switch. Be aware that enabling debug could flood the receiving Telnet session or asynchronous port with raw data.

The **debug** parameter specifies which debug mode to enable. The following table describes available options.

| Debug Options | Description                                          |
|---------------|------------------------------------------------------|
| DMA           | Operations related to Direct Memory Access requests. |
| QOS           | Operations related to Quality of Service.            |
| PHY           | Operations related to the PHY port interfaces.       |
| ALL           | All debug options.                                   |

When the **output** parameter is set to **console**, debugging information is sent to the console. By default, data goes to the port where the **enable switch debug** command was received. Use this option if the command is used in a script, since a script is not received on a port.

The **timeout** parameter specifies the time in seconds for which any switch debugging is enabled. This reduces the risk of the switch and the display being overloaded with too much debugging information. This value overrides any previous switch debugging timeout values, even if they were specified for other debugging modes. If **timeout** is not specified, the time out is the most recent **timeout** value previously used in an **enable vlan debug** command, or **none** if it has not been previously set.

**Example** To enable the ARL switch debugging mode, use the command:

```
ena swi deb=arl
```

**Related Commands** [disable switch debug](#)  
[show switch](#)  
[disable debug active](#) in Chapter 4, Configuring and Monitoring the System  
[show debug active](#) in Chapter 4, Configuring and Monitoring the System

## enable switch hash

**Syntax** ENABle SWITch HASH={L2 | L3 | L4} [, ...]

**Description** This command enables the use of packet fields in the Link Aggregate function. It applies to bridged traffic only.

The Link Aggregate hash function determines the member port of a trunk group when the packet is destined to a port within a trunk group. The Link Aggregate hash function also determines the flow group to which a packet belongs when the packet is associated with a traffic class that has flow groups enabled.

When Layer 4 hashing is enabled Layer 3 header information is also used to hash traffic to a trunked port.

When traffic is routed at Layer 3 to a trunk group, only Layer 3 header information is used to hash traffic to the ports in the trunk group, irrespective of whether Layer 2, 3, or 4 hashing is enabled.

The **hash** parameter specifies the type of packet fields to use in the Link Aggregate hash function, as a comma-separated list. The default is for all fields to be used. The following table describes individual options.

| Value | Packet Field                                                                                                                       |
|-------|------------------------------------------------------------------------------------------------------------------------------------|
| L2    | Source and destination MAC addresses.                                                                                              |
| L3    | Source and destination IP addresses<br>Destination IPX network addresses                                                           |
| L4    | Source and destination TCP or UDP ports for TCP/IP and UDP/IP packets<br>Source and destination IPX socket numbers for IPX packets |

**Examples** To enable the use of Layer 4 fields in a packet in the Link Aggregate hash function, use the command:

```
ena swi hash=L4
```

To enable the use of Layer 2 and Layer 3 fields in a packet in the Link Aggregate hash function, use the command:

```
ena swi hash=L2,L3
```

**Related Commands** [disable switch hash](#)  
[show switch](#)

## enable switch learning

---

**Syntax** ENABle SWItch LEarning

**Description** This command enables the dynamic learning and updating of the forwarding database. Learning is enabled by default.

**Example** To enable the switch learning function, use the command:

```
ena swi le
```

**Related Commands** [disable switch learning](#)  
[show switch](#)

## enable switch mclimiting

---

**Syntax** ENABle SWItch MClimiting

**Description** This command enables multicast packet limiting on the switch.

The rate at which multicast packets are limited is set with the **set switch port** command. The **bclimit** option sets the maximum data rate of reception for both multicast and broadcast packets as independent limits for multicast and broadcast packets cannot be set.

This command has no effect unless at least one of the switch's ports has broadcast rate limiting enabled. When multicast rate limiting is enabled, then it is active for all ports that:

- already have broadcast limiting set and enabled
- subsequently have broadcast limiting enabled after multicast limiting is enabled

**Examples** To enable multicast packet limiting on the switch, use the command:

```
ena swi mc
```

**Related Commands** [disable switch mclimiting](#)  
[set switch port](#)

## enable switch mirror

---

**Syntax** ENABle SWItch MIRRor

**Description** This command first checks that the number of egress ports and the number of ingress ports to be mirrored are not greater than their maximums. If either one is greater, a message is given and mirroring is not enabled. Once traffic mirroring on the switch is enabled, mirrored traffic is sent, as long as a valid mirror port is defined and sources of mirror traffic have been configured. The default state of mirroring is disabled.

**Important** Four or more ports set to mirror traffic to the mirror port may significantly reduce switch performance.

If a packet is Layer 3 switched and mirrored, then the packet is always transmitted from the mirror port with a VLAN tag.

**Example** To enable traffic mirroring, use the command:

```
ena swi mirr
```

**Related Commands**

- [disable switch mirror](#)
- [set switch mirror](#)
- [set switch port](#)
- [show switch](#)
- [show switch port](#)

## enable switch port

---

**Syntax** ENABle SWitCh Port={*port-list*|ALL} [AUTOmDi]  
[EGResSqueue=*queue-list*] [FLOw=PAUSE|JAMMING]

where

- *port-list* is a single port number or a group as either a comma-separated list, range (specified as *n-m*), or a combination of the two. Port numbers start at 1 and end at *m*, where *m* is the highest numbered switch Ethernet port.
- *queue-list* is an egress queue number, a range of queue numbers (specified as *n-m*), or a comma separated list of queue numbers and/or ranges. Egress queue numbers start at 0 and end at 7.

**Description** This command enables one or more of:

- a port or group of ports on the switch (switch ports are enabled by default)
- auto MDI/MDI-X mode on the ports
- flow control on the ports
- egress queues on the ports

When you enable a port, it starts to send or receive packets. Note that enabling a port does not enable STP operation on the port.

To enable a port that has been disabled by the Port Security function, use the **set switch port** command rather than this command.

The **port** parameter specifies the port to enable.

The **flow** parameter specifies the type of flow control to be enabled for the port. If **pause** is specified, flow control for full duplex ports by sending PAUSE frames will be enabled. If **jamming** is enabled, flow control for half duplex by sending a jamming signal will be enabled. No flow control is enabled by default.

The **automdi** parameter enables auto MDI/MDI-X mode, and sets the polarity to the default of MDI-X. Auto MDI/MDI-X mode is enabled by default.

The **egressqueue** parameter specifies the egress queue(s) that are to be enabled. If you specify this parameter, you must supply a value for it. At switch startup all egress queues are enabled by default.

**Examples** To enable ports 2, 4 and 6, use the command:

```
ena swi po=2,4,6
```

To enable auto MDI/MDI-X on ports 2 and 4, use this command:

```
ena swi po=2,4 autom
```

To enable egress queues 0, and 3 to 5 on port 1, use the command:

```
ena swi po=1 egr=0,3-5
```

To enable pause frames on port 1, use the command:

```
ena swi po=1 flo=pause
```

**Related Commands** [disable switch port](#)  
[set switch port](#)  
[show switch port](#)

---

## enable switch port vlan

---

**Syntax**    ENable SWitch Port={*port-list*|ALL} VLAN[={*vlan-name*|1..4094|ALL}]

where:

- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.
- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen.

**Description**    This command enables one or more ports for VLANs to which they belong. A port is automatically enabled for a VLAN when it is added to that VLAN, however, it can be disabled using the [disable switch port vlan](#) command, or automatically disabled by thrash limiting or QoS protection.

The **port** parameter specifies the port or ports to enable. If a trunked port is specified, all ports in the trunk are enabled.

The **vlan** parameter specifies the VLAN or VLANs for which ports are enabled. Specified ports must be a member of the VLAN. If no value or **all** is specified, the specified ports are enabled for all VLANs to which they belong.

Note that when a disabled VLAN is re-enabled on a port, the port automatically has ingress filtering disabled, as long as there are no other VLANs disabled on the port, and as long as ingress filtering was not previously enabled by using the **set switch port** command.

**Examples**    To enable the default vlan on port 1, use the command:

```
ena swi po=1 vlan=1
```

**Related Commands**    [disable switch port vlan](#)  
                          [set switch port](#)  
                          [show switch port](#)

## enable vlan debug

---

**Syntax**    ENABle VLAN={vlan-name|1..4094|ALL} DEBUg={PKT|ALL}  
             [OUTput=CONSOLE] [TIMEOut={1..4000000000|NONE}]

where *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.

**Description**    This command enables debugging options for the specified VLAN or all VLANs. Be aware that enabling debug may flood the receiving Telnet session or asynchronous port with raw data. The default is for all VLAN debugging to be disabled.

The **debug** parameter specifies which debugging mode is enabled. If **pkt** is specified, packet debug mode (displaying raw ASCII packets) is enabled. If **all** is specified, all debugging is enabled.

The **output** parameter set to **console** specifies that the debugging information produced is sent to the console. The debugging data is by default sent to the port on which it received the **enable vlan debug** command. Use this option if the command is used in a script, since a script is not received on a port.

The **timeout** parameter specifies the time in seconds when debugging is enabled on a specific VLAN. This reduces the risk of the switch and the display being overloaded with too much debugging information. This value overrides previous VLAN debugging timeout values for the VLAN, even if they were specified for other debugging modes. If **timeout** is not specified, the most recent time out value for the **enable vlan debug** command is used, or **none** if none had been set.

**Example**    To enable all debugging on the *marketing* VLAN, use the command:

```
ena vlan=marketing deb=all
```

**Related Commands**    [disable vlan debug](#)  
                          [show vlan port](#)

## purge lacp

---

**Syntax**    PURge LACP

**Description**    This command returns LACP to the status that existed when first powered on. It destroys all LACP configurations and restores the defaults to the configurable parameters. LACP parameters for all ports are reset to their defaults.

**Example**    To purge the LACP configuration, use the command:

```
pur lacp
```

**Related Commands**    [enable lacp](#)  
                          [disable lacp](#)  
                          [set lacp port](#)



## reset lacp port counter

---

**Syntax** RESET LACP Port [= {*port-list* | ALL}] COUNTER

where *port-list* is a port number, range (specified as *n-m*), or comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.

**Description** This command resets all LACP counters for the specified switch ports.

The **port** parameter specifies the ports. If **all** is specified, all port counters in the switch are reset. The default is **all**.

**Examples** To reset the LACP counters for all ports, use the command:

```
reset lacp po cou
```

**Related Commands** [purge lacp](#)  
[show lacp port](#)

## reset switch

---

**Syntax** RESET SWITCH

**Description** This command resets the switch. All dynamic switch information is cleared. All ports are reset. All counters and timers are reset to zero.

**Example** To reset the switch, use the command:

```
reset swi
```

**Related Commands** [show switch](#)  
[show switch filter](#)

## reset switch accelerator counter

---

**Syntax** RESET SWITCH ACCELERATOR COUNTER [= {ALL | DEFAULT | FAB | MAC | MIB}]

**Description** This command clears the counters stored in the IPv6 accelerator card on AT-8948 switches.

The **counter** parameter specifies which block of counters to clear. If **all** is specified, all counters are cleared. If **fab** is specified, the internal IPv6 accelerator card fabric interface counters are cleared. If **mac** is specified, the internal IPv6 accelerator card MAC counters are cleared. If **mib** is specified, mode-specific SNMP counters are cleared. The default is **default**, which clears the MAC and MIB counters.

**Examples** To clear all IPv6 accelerator card counters, use the command:

```
reset swi accel cou=all
```

**Related Commands** [show ipv6 counter](#) in Chapter 23, Internet Protocol version 6 (IPv6)

## reset switch port

---

**Syntax**    RESET SWitch Port={*port-list*|ALL} [COUnter]

where *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.

**Description**    This command resets a port or group of ports on the switch. All packets queued for reception or transmission on the port are discarded, the port is reset at the hardware level to the configured speed and duplex mode, and autonegotiation of speed and duplex mode is activated. Switch port counters are reset to zero. This command clears packets stuck in a queue, perhaps after a packet storm.

The **port** parameter specifies the ports to be reset.

The **counter** parameter specifies that only switch port counters are reset. If this parameter is not used, the switch port is fully reset.

**Example**    To reset port 3, use the command:

```
reset swi po=3
```

**Related Commands**    [disable switch port](#)  
                          [enable switch port](#)  
                          [show switch port](#)

## set lacp port

**Syntax** SET LACP Port=[*port-list*/ALL] [Adminkey=*key-number*]  
[PRIOrity=*priority*] [MODe={ACTIve|PASSive}]  
[PERIodic={FAST|SLOW}]

where:

- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.
- *key-number* is a integer from 0 to 65535
- *priority* is a integer from 0 to 65535

**Description** This command modifies the value of parameters for LACP ports.

The **port** parameter specifies the ports for which parameters are modified. If the command would succeed on a subset of the ports specified, but cause an error on the others, then the command as a whole fails and has no effect. Reference in the descriptions below to an individual port should be taken as a reference to all ports selected by the **port** parameter.

The **adminkey** parameter specifies the Admin LACP port key. This affects the LACP port key that is generated but does not determine its value. You can use this parameter to prevent ports from being aggregated when they might otherwise form a trunk. By default all ports that can be aggregated are given the same LACP port key. The default for **adminkey** is 1.

The **priority** parameter specifies the LACP port priority. This value is used to decide which ports should be selected when being added to a trunk group (where there are more links existing between the two devices than the switch is able to aggregate). The default is one. This means that port number governs which ports are selected (low port number equals high priority). Excess ports are put into a standby mode. In this mode they remain untrunked, but still able replace a link that goes down.

The **mode** parameter specifies whether the port runs in LACP passive or active mode. A port in passive mode sends an LACPDU in response to receiving one; whereas, a port in active mode sends LACPDU at regular intervals as specified by the **periodic** parameter.

The **periodic** parameter specifies the rate at which the LACP port transmits updates. A port in fast mode transmits one LACPDU every second; a port in slow mode transmits one LACPDU every thirty seconds.

**Related Commands**

- [delete lacp port](#)
- [add lacp port](#)
- [show lacp port](#)

## set lacp

**Syntax** SET LACP PRIOrity=*priority* [THRASHAction={LEarndisable|LINKdown|NONE|PORtdisable|VLANDisable}][THRASHTimeout={None|1..86400}]

where *priority* is an integer from 0 to 65535

**Description** This command modifies the LACP parameters.

The **priority** parameter specifies a numeric value that is used as part of the system priority calculation. When systems with multiple links connect and use LACP to control link aggregation, each system compares its system priority data identifiers to determine which system should control the links. A system identifier comprises a system priority component (configured by this parameter) followed the system's MAC address. Link control is assigned to the system with the numerically *lower* system priority data identifier. The default is 32768.

The **thrashaction** parameter specifies the action the switch takes when it detects MAC address thrashing on any trunk created by LACP. Thrashing occurs when one or more ports or trunks repeatedly learn the same MAC addresses, for example, as a result of a network loop. The switch applies the trunk's **thrashaction** to all ports in the trunk.

Take care with the **thrashaction** parameter because misuse can impair your network operation. Set the **thrashaction** parameter to:

- **none** to apply no thrash limiting on the trunk.
- **learndisable** to disable MAC address learning on all ports in the thrashing trunk, until the period specified with the **thrashtimeout** parameter has elapsed. The default is **learndisable**.
- **portdisable** or **linkdown** to disable all ports in the thrashing trunk until either the period specified by the **thrashtimeout** parameter has elapsed, or until the ports or subset of ports in the trunk are re-enabled by the **enable switch port** command. If you specify **linkdown**, the link state is down; if you specify **portdisable**, the link state remains up.
- **vlandisable** to block all traffic on the VLAN where the address was learned, on all ports in the thrashing trunk, until either the period specified by **thrashtimeout** has elapsed, or until the ports are re-enabled using the **enable switch port vlan** command. When **thrashaction=vlandisable**, there is only one timer per trunk, so if multiple VLANs have been disabled on a trunk, the timer starts when the last VLAN was disabled. When the timer expires, all VLANs are re-enabled on the trunk. When **thrashaction=vlandisable**, ingress filtering is automatically enabled on all ports in the trunk.

The **thrashtimeout** parameter specifies the time, in seconds, for which the switch employs the thrash action specified by the **thrashaction** parameter. The **thrashtimeout** cannot be set to **none** if **thrashaction=learndisable**. If **thrashtimeout=none**, and **thrashaction** is then changed to **learndisable**, then the switch automatically changes the **thrashtimeout** to 1 second.

If **none** is specified, the trunk is not automatically re-enabled, but individual ports can be re-enabled by using the **enable switch port** command for **thrashaction=portdisable** or **linkdisable**, and the **enable switch port vlan** command for **thrashaction=vlandisable**. The default is 1 second.

**Examples** System A is to connect to system B using LACP and System B is to control their aggregated links.

System A has a MAC address of 00-00-cd-00-0d-42 and has been assigned an LACP priority value of 500. System B has a MAC address of 00-00-cd-00-0d-52.

In order to ensure that System B controls the links, its LACP priority must be set to a value **lower** than 500. The LACP priority on System B is therefore set to 300. Note that system control is determined by the values set by the LACP priority values because these have a greater numeric significance than MAC Addresses.

```
set lacp prio=300
```

**Related Commands** [show lacp](#)

---

## set switch ageingtimer

---

**Syntax** SET SWITCh AGEingtimer=10..630

**Description** This command sets the threshold of the ageing timer in seconds, after which a dynamic entry in the forwarding database is automatically removed.

The maximum setting of 630 seconds is 10 minutes 30 seconds. The default is 300 seconds (5 minutes). The ageing timer is rounded down to the nearest multiple of 10.

**Example** To set the ageing timer to 180 seconds (3 minutes), use the command:

```
set swi age=180
```

**Related Commands** [disable switch ageingtimer](#)  
[enable switch ageingtimer](#)  
[show switch](#)

---

## set switch cputxpriorityoverride

---

**Syntax** SET SWITCh CPUTXPriorityoverride={NONE|0..7}

**Description** This command specifies the 802.1p value to be inserted into packets that the CPU sends to the switching chip to be transmitted.

The **cputxpriorityoverride** parameter specifies the value to put into the 802.1p field of tagged packets being sent from the CPU. The default of the parameter is **none**, where the CPU does not set the 802.1p field in packet headers. **none** can be specified to stop the overriding.

**Examples** To set the switch to place the value 7 in the 802.1p field of packets sent from the CPU, use the command:

```
set swi cputxp=7
```

To disable overriding CPU transmit priority, use the command:

```
set swi cputxp=none
```

**Related Commands** [set ip dscpoverride](#)  
[set switch cputxqueueoverride](#)

## set switch cputxqueueoverride

---

**Syntax** SET SWITCh CPUTXQueueoverride={NONE|0..7}

**Description** This command specifies the transmit queue to be used for packets that are sent from the CPU to the switching chip for transmission.

The **cputxqueueoverride** parameter specifies the egress queue into which the switching chip places packets from the CPU. The default of the parameter is **none**, which means that packets from the CPU are put into egress queue 0. Also specify **none** to stop the overriding.

**Examples** To set the switch to put CPU-initiated packets into egress queue 7, use the command:

```
set swi cputxq=7
```

To stop specifying an egress queue for CPU-initiated packets, use the command:

```
set swi cputxq=none
```

**Related Commands** [set ip dscpoverride](#)  
[set switch cputxpriorityoverride](#)

## set switch dlflimit

---

**Syntax** SET SWITCh DLFLimit={NONE|*limit*}

where *limit* is a decimal number, between 100 and 95300 in Kbytes per second

**Description** This command sets the storm protection limit for unknown packets on the switch.

The **dlflimit** parameter specifies the maximum data rate at which destination lookup failure packets are received by the switch per second. Packets beyond this limit are discarded. If **none** is specified, destination lookup failure protection is implicitly disabled on the switch. The maximum rate is 95300 Kbytes per second.

**Examples** To set the limit of reception of destination lookup failure packets to 1000 Kbytes per second, use the command:

```
set swi dlfl=1000
```

To disable the rate limiting of destination lookup failure packets on the switch, use the command.

```
set swi dlfl=none
```

**Related Commands** [show switch](#)

---

## set switch enhancedmode

---

**Syntax** SET SWItch ENHancedmode={QOSCounters|NEXThop|NONE}

**Description** This command rearranges memory in the switch so that it can store:

- QoS traffic class counters
- a greater number of routes than usual

If you specify **qoscounters**, the switch monitors QoS counters for traffic classes. Note that the maximum number of traffic classes is reduced. This option must be set before you configure QoS storm protection (see [“Storm Protection” on page 27-12 of Chapter 27, Quality of Service \(QoS\)](#)).

The **qoscounters** parameter is dynamically added to a text file called switch.ini, which is read at an early stage of the startup process. For the enhanced mode to take effect, you must restart the switch after entering this command. The switch remains in enhanced mode until you change it with a command or you delete the switch.ini file.

An alternative to this command is to load a switch.ini file into the file system that contains the appropriate string. This could be useful when deploying configurations on many switches.

If you specify **nexthop**, the switch stores up to 5000 individual routes (nexthops). The maximum number of multicast groups and traffic classes is reduced.

**Examples** To turn on the monitoring of QoS traffic class counters, use the command:

```
set swi enh=qosc
```

To turn off the monitoring of QoS traffic class counters, use the command:

```
set swi enh=none
```

**Related Commands** [show qos port counters](#)  
[show switch](#)

## set switch hwlearndelay

**Syntax** SET SWItch HWLearndelay=0..100000

**Description** This command sets the length of time, in milliseconds, between activity in the IP route learning system and the beginning of hardware route learning. The default value is 4.

In environments with a very high number of IP routes in use, increasing this delay to several seconds allows you to prioritise software route processing over hardware route learning, which improves the software route convergence time. It also increases the latency of hardware route learning. Departure from the default setting is not recommended for most network systems, and may impact the routing and CPU performance of a device.

When the delay is non-zero, the switch places routes in a queue for adding to its hardware table. You can set the length of the queue by using the [set switch hwrouteupdate](#) command on page 7-104.

**Example** To force the switch to wait 5 seconds after the last IP route update before updating hardware routing, use the command:

```
set swi hwl=5000
```

**Related Commands** [set switch hwrouteupdate](#)  
[show switch](#)

## set switch hwrouteupdate

**Syntax** SET SWItch HWRouteupdate=1..*maximum*

**Description** This command sets the length of the hardware route update queue. The switch uses the queue when hardware learning delay is enabled (the default situation). The switch learns new routes in software, then places them into the queue for adding to its hardware routing table.

The **hwrouteupdate** parameter specifies the maximum possible number of entries in the queue. The *maximum* and default values depend on the amount of memory on the switch, as shown in the following table:

| Memory Size (Mbytes) | Default length<br>(number of entries) | Maximum possible length<br>(number of entries) |
|----------------------|---------------------------------------|------------------------------------------------|
| up to 128            | 200000                                | 200000                                         |
| 129-256              | 1000000                               | 1500000                                        |
| more than 256        | 3000000                               | 4000000                                        |

**Example** To make the queue as long as possible on a switch with 256 Mbytes of memory, use the command:

```
set swi hwr=4000000
```

**Related Commands** [set switch hwlearndelay](#)  
[show switch](#)



## set switch jumbo

**Syntax** SET SWITCh JUmbO={ON|OFF}

**Description** This command sets the maximum packet size switch ports can receive.

| Parameter  | Description                                                                                                                                                     |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Jumbo      | Type of packet.<br>Default: <b>off</b>                                                                                                                          |
| <b>on</b>  | Maximum size for ports that work at speeds of either 10Mbps or 100Mbps is 9710 bytes.<br>Maximum size for ports that work at speeds of 1000Mbps is 10240 bytes. |
| <b>off</b> | Maximum packet size for all switch ports is 1522 bytes.                                                                                                         |

**Example** To enable the switch to accept jumbo frames, use the command:

```
set swi ju=on
```

**Related Commands** [show switch](#)

## set switch mirror

**Syntax** SET SWITCh MIRRor={NONE|*port*}

where *port* is a single switch port number. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port.

**Description** This command sets the mirror port for the switch and removes it from the default VLAN. If another port was previously set as the mirror port, this command returns it to the default VLAN as an untagged port. The mirror port is the port to which all mirrored traffic is sent. The source of mirror traffic is configured with the **set switch port** command.

The **mirror** parameter specifies the switch port to which mirror traffic is to be sent. The specified port must belong only to the default VLAN, either as an untagged port or a tagged port. The port cannot be part of a trunk group. If the value **none** is specified, then no mirror port is defined for the switch and mirroring is disabled. The mirror port cannot be added to VLANs.

If a packet is Layer 3 switched and mirrored, then it is always transmitted from the mirror port with a VLAN tag.

**Example** To set the mirror port to port 12, use the command:

```
set swi mirr=12
```

**Related Commands** [disable switch mirror](#)  
[enable switch mirror](#)  
[set switch port](#)  
[show switch](#)  
[show switch port](#)

## set switch nestedtpid

**Syntax** SET SWITCh NESTedtpid=*tag-number*

where *tag-number* is a two byte hexadecimal number

**Description** This command sets the Tag Protocol Identifier (TPID) for packets transmitted out of CORE ports on a nested VLAN. The nested VLAN feature is also known as *VLAN double tagging*.

The **nestedtpid** parameter specifies the Ethernet type of the tagged packet. This is set to 0x8100 by default when a nested VLAN is created.

This command specifies the TPID value that applies to all VLANs used for nested VLANs. The TPID value cannot be set for only one particular nested VLAN if more than one nested VLAN is created.

**Example** To set the **nestedtpid** value to 0x8800, use the command:

```
set swi nest=0x8800
```

**Related Commands** [add vlan port](#)  
[create vlan](#)  
[delete vlan port](#)  
[show vlan port](#)

## set switch port

**Syntax** SET SWITCh PORT={*port-list*|ALL} [ACCEptable={ALL|VLAN}]  
 [BCLimit={NONE|*limit*} [DESCRiption=[*description*]]  
 [EGRESSlimit={*bandwidth*|DEFAULT}] [IGMPAction={DENY|  
 REPLACE}] [IGMPFilter={NONE|*filter-id*}]  
 [IGMPMAxgroup={NONE|1..65535}] [INFILTering={OFF|ON}]  
 [INTRusionaction={DISAbLe|DISCard|TRAp}] [LEARn={NONE|  
 0|1..256}] [MIRROR={BOTH|NONE|RX|TX}]  
 [MODE={AUTOnegotiate|MASTER|SLAve}] [POLarity={MDI|  
 MDIX}] [RELearn={OFF|ON}] [SPEED={AUTOnegotiate|  
 10MAUTO|10MHAlf|10MFULL|10MHAUTO|10MFAuto|100MAUTO|  
 100MHAlf|100MFULL|100MHAUTO|100MFAuto|1000MHAlf|  
 1000MFULL|1000MFAUTO}] [THRASHAction={LEArndisable|  
 LINKDown|NONE|PORTdisable|VLANdisable}]  
 [THRASHTimeout={None|1..86400}] [VLANSTATustrap={ON|  
 OFF}]

where:

- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port.
- *description* is a string 1 to 47 characters long. Valid characters are any printable characters.
- *bandwidth* is the maximum bandwidth available to the port in kbps, specified in multiples of approximately 650kbps.
- *limit* is a decimal number between 100 and 95300, in Kbytes per second
- *filter-id* is a decimal number in the range 1 to 99.

**Description** This command modifies the value of parameters for switch ports.

The **port** parameter specifies the ports for which parameters are modified. If the command succeeds on a subset of the ports specified but causes an error on the others, then the command as a whole fails and has no effect. Reference in the descriptions below to an individual port should be taken as a reference to all ports selected by the **port** parameter.

The **acceptable** parameter sets the Acceptable Frame Types parameter in the Ingress Rules, which controls reception of VLAN-tagged and VLAN-untagged frames on the port. If **all** is specified, the Acceptable Frame Types parameter is set to Admit All Frames. If **vlan** is specified, the parameter is set to Admit Only VLAN-tagged Frames, and any frame received that carries a null VLAN Identifier (VID) is discarded by the Ingress Rules. Untagged frames and priority-tagged frames carry a null VID. Untagged frames admitted according to the **acceptable** parameter have the VID of the VLAN for which the port is untagged associated with them. The **acceptable** parameter can only be set if the port is untagged for one VLAN. In this case, the default is **all**, admitting all tagged and untagged frames. If the port is tagged for all the VLANs to which it belongs, the **acceptable** parameter is automatically set to VLAN, and cannot be changed to admit untagged frames.

The **bclimit** parameter specifies the maximum data rate of reception of Layer 2 broadcast packets by the specified port, in Kbytes per second. Specified packets beyond this limit are discarded. The **bclimit** parameter specifies the maximum rate for both broadcast and multicast rate limiting on the specified port, or ports. It is not possible to set independent limits for multicast or broadcast packets. In summary, you can enable the following on a per port basis:

- broadcast rate limiting
- broadcast and multicast rate limiting

To enable the limiting of multicast packets, use the [enable switch mclimiting command on page 7-92](#). If **none** or **0** is specified, broadcast packet limiting is implicitly disabled on the port. If any other value is specified, the reception of broadcast or multicast packets is limited to that number of packets per second. The latest parameter entered supersedes earlier values. The default is **off**.

The **description** parameter describes the port. It is displayed by the [show switch port](#) and sets the value of the ifDescr MIB object, but does not affect the operation of the switch. You can also enter the parameter without a value to remove an existing description. The default is no description.

The **egresslimit** parameter specifies the maximum bandwidth (in Kbps) available to the port. If you specify a value less than 1 Gbps, the value is rounded up to the nearest 648 Kbps. If you specify a value greater than 1 Gbps, the value is rounded up to the nearest 41.5 Mbps. This parameter does not set the exact bandwidth of the port; it determines the rate at which data leaves internal queues prior to being transmitted onto the line. Note that because the frames' header and trailer information (that encapsulate the data) are not included in the egress limit calculation, their addition adds an overhead to the overall port transmission bandwidth. [Figure 7-15](#) shows how the frame size impacts upon the port's actual data transmission rate (bandwidth) onto the line. The larger the frame size, the closer the actual percentage of bandwidth on line gets to the **egresslimit** that has been set. For example, if port 1 is a Gigabit port, using the **set switchport=1 egresslimit=500000** command does not limit the port to transmitting 500000 Kbps onto the line. The default is the maximum bandwidth.

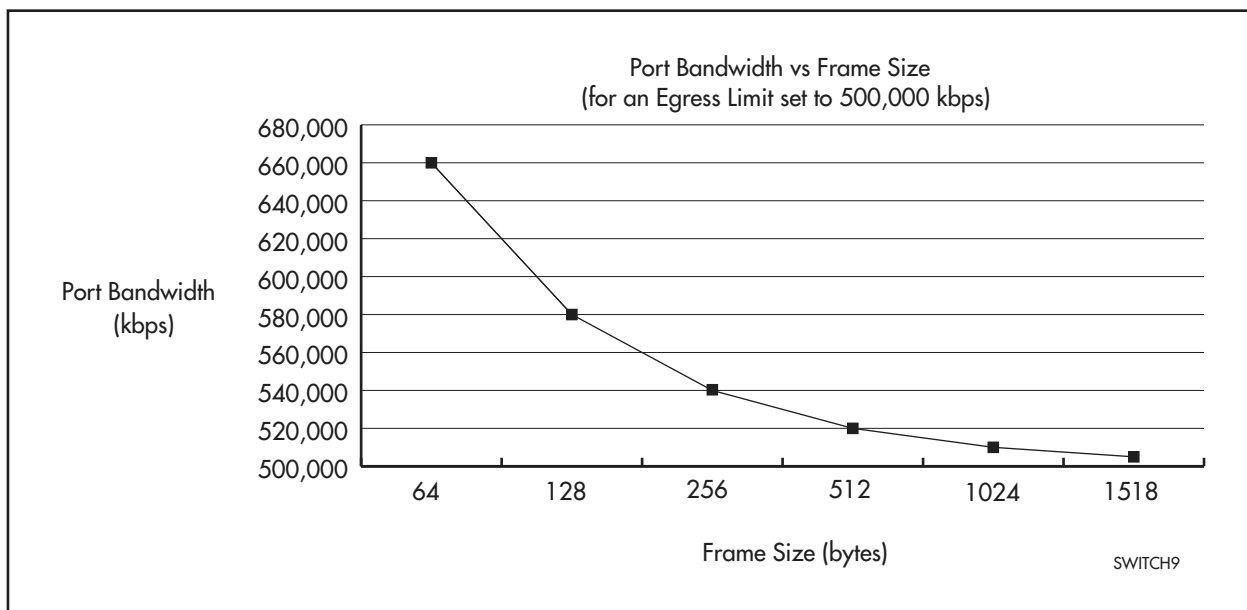
If the maximum bandwidth has not been set and the user changes the speed of the port, then the maximum bandwidth for the port is automatically

recalculated. If the user sets this parameter to a particular value and then increases the speed of the port, the maximum bandwidth remains unchanged. If **default** is specified, then the switch automatically recalculates the maximum bandwidth based upon the current speed of the port.

Because a port can be associated with a trunk group that comprises two or more ports, the maximum bandwidth can exceed the (common) speed of each individual port within the group. Also, the maximum bandwidth for the whole trunk group is determined by the **egresslimit** that is set for the trunk group's master port. For this reason we recommend setting the **egresslimit** to a common value for all ports within a particular trunk group.

If the value of the **egresslimit** parameter is set to anything other than the default and a QoS policy is assigned to the port, the maximum bandwidth of any traffic class in the policy is restricted to that maximum bandwidth.

Figure 7-15: Bandwidth for various frame sizes



The **igmpaction** parameter specifies the action to take when the number of multicast group memberships associated with the port reaches the limit set by **igmpmaxgroup**. If you specify **deny**, then additional Membership Reports are discarded until existing group memberships age out. If you specify **replace**, then additional membership entries will replace existing membership entries. The default is **deny**.

The **igmpfilter** parameter specifies the number of an IGMP filter to apply to the port. An IGMP filter controls the multicast groups that the port can be a member of by filtering IGMP Membership Reports from hosts attached to the port. If you specify a filter number, an IGMP filter with the specified number must already exist. You can apply an IGMP filter to more than one switch port, but a single port can have only one filter assigned to it. Specify **none** to apply no filter to the port, or to remove an existing filter from the port. The default is **none**.

The **igmpmaxgroup** parameter specifies the maximum number of multicast groups that the port can join. Specify **none** to set no limit. The default is **none**. For trunk ports, the value of **igmpaction**, **igmpfilter**, and **igmpmaxgroup** for the master port will apply to the trunk.

The **infiltering** parameter enables or disables Ingress Filtering of frames admitted according to the **acceptable** parameter, on the specified ports. Each port on the switch belongs to one or more VLANs. If **infiltering** is set to **on**, Ingress Filtering is enabled: any frame received on a specified port is only admitted if the port belongs to the VLAN with which the frame is associated. Conversely, any frame received on the port is discarded if the port does not belong to the VLAN with which the frame is associated. Untagged frames admitted by the **acceptable** parameter are admitted, since they have the numerical VLAN Identifier (VID) of the VLAN for which the port is an untagged member. If **off** is specified, Ingress Filtering is disabled, and no frames are discarded by this part of the Ingress Rules. The default setting is **off**.

The **intrusionaction** parameter specifies the action taken when the port receives packets from addresses that are not part of the learned list of addresses as specified by the **learn** parameter. If **discard** is specified, packets received from MAC addresses not on the port's learn list are forwarded to other ports in the appropriate VLAN; however, return traffic to MAC addresses not on the port's learn list are discarded. If **trap** is specified, packets received from MAC addresses not on the port's learn list are forwarded to other ports in the appropriate VLAN and an SNMP trap is generated. However, return traffic to MAC addresses not on the port's learn list are discarded. If **disable** is specified, the first time a packet is received from a MAC address not on the port's learn list and is forwarded to other ports in the appropriate VLAN, an SNMP trap is generated and the port is disabled. The previous process occurs within one second of the first packet arriving on the port that is not on the port's learn list.

During the one second interval after the first packet not on the port's learn list has been received, it may be possible for packets that are not on the port's learn list to be switched between ports in the VLAN.

To re-enable the port, disable the Port Security function on the port. The default is **discard**.

The **learn** parameter specifies whether the security feature of limiting the number of MAC addresses learned on this port is enabled. If **none** or zero is specified, all MAC addresses are learned on this port and the Port Security function is disabled. If the port was disabled by the Port Security function, it is re-enabled. If a number from 1 to 256 is specified for the **learn** parameter, the switch stops learning MAC addresses on this port once the number of MAC addresses has been reached and the port is locked. If the **learn** parameter is set to a value lower than the number of MAC addresses currently learned, then the port is unlocked if previously locked, learned MAC addresses are cleared from the forwarding database for the port, and learning restarts. Packets received from other addresses after this time are handled as intrusion packets (see the **intrusionaction** parameter). The default is **none**.

The **relearn** parameter determines whether dynamic or static MAC address learning is to be used on this port. This parameter has no effect when the security feature limiting the number of MAC addresses is disabled (i.e. when **learn=0** or **none**). If the **relearn** parameter is **off**, then static MAC address learning are used. Once a MAC address has been learned, it remains permanently in the learning database. If the **relearn** parameter is **on**, then dynamic MAC address learning is used. If a MAC address is unused for a period of time, then it is removed from the learning database. Either another or the same, MAC address can be learned and stored in the vacant position in the learning database. When **relearn** is enabled on a port, entries in the learning database are removed. The elapsed time before a MAC address entry is removed can be set with the **set switch ageingtimer** command. The default is **off**.

The **mirror** parameter specifies the role of these ports as a source of mirror traffic. If **rx**, **tx**, or **both** are specified, the ports specified in the **port** parameter are checked to ensure that the number of mirrored ports does not exceed the threshold. Ports are not set if this check is not passed. If **none** is specified, no traffic received or sent on these ports is mirrored. If **rx** is specified, all traffic received on these ports is mirrored. If **tx** is specified, all traffic transmitted on these ports is mirrored. If **both** is specified, all traffic received and transmitted is mirrored. Traffic is mirrored when a mirror port is defined and mirroring is enabled. The default is **none**.

**Important** Four or more ports set to mirror traffic to the mirror port may significantly reduce switch performance.

The **mode** parameter applies to gigabit interfaces and may be used to force the interface to operate in master or slave mode. This is not normally required and should be used only when the link partner does not support autonegotiation of master/slave mode. When mode is set to **master** or **slave**, the interface is forced to operate in the specified mode. The default is **autonegotiate**.

The **polarity** parameter specifies a fixed MDI or MDI-X polarity for fixed switch ports with copper interfaces. The configured **polarity** parameter value is not retained if auto MDI/MDI-X is enabled and disabled. It is always set to the default when auto MDI/MDI-X is disabled. The default is MDI-X. The **polarity** parameter is not valid for SFP transceivers plugged into the SFP sockets.

The **speed** parameter specifies the configured line speed and duplex mode of the port. For the options supported on each type of port, see [“Port types and speed” on page 7-10](#). If **autonegotiate** is specified, the port autonegotiates the highest mutually possible line speed and duplex mode with the link partner. If **10Mauto** or **100Mauto** is specified, the port autonegotiates with the link partner to determine duplex mode but accepts operation only at the specified speed. If **10Mfauto**, **10Mhauto**, **100Mfauto**, **100Mhauto**, or **1000Mfauto** is specified, the port autonegotiates with the link partner but accepts operation only at the specified speed and duplex mode. If **10Mhalf**, **10Mfull**, **100Mhalf**, **100Mfull**, **1000Mhalf**, or **1000Mfull** is specified, then autonegotiation is disabled and the interface is forced to operate at the specified speed and duplex mode, regardless of whether the link partner is capable of working at that speed. The default is **autonegotiate**. The following table describes switch port speeds.

| Option     | Meaning                              |
|------------|--------------------------------------|
| 10Mauto    | 10Mbps, autonegotiate duplex mode    |
| 10Mhalf    | 10Mbps, half duplex, fixed           |
| 10Mfull    | 10Mbps, full duplex, fixed           |
| 10Mhauto   | 10Mbps, half duplex, autonegotiate   |
| 10Mfauto   | 10Mbps, full duplex, autonegotiate   |
| 100Mauto   | 100Mbps, autonegotiate duplex mode   |
| 100Mhalf   | 100Mbps, half duplex, fixed          |
| 100Mfull   | 100Mbps, full duplex, fixed          |
| 100Mhauto  | 100Mbps, half duplex, autonegotiate  |
| 100Mfauto  | 100Mbps, full duplex, autonegotiate  |
| 1000Mhalf  | 1000Mbps, half duplex, fixed         |
| 1000Mfull  | 1000Mbps, full duplex, fixed         |
| 1000Mhauto | 1000Mbps, half duplex, autonegotiate |
| 1000Mfauto | 1000Mbps, full duplex, autonegotiate |

When a port is included in a trunk group, it must operate at the speed specified for the trunk group and in full duplex mode. This speed is selected by autonegotiation with the link partner. If the port is removed from the trunk group, the previously configured speed and duplex mode are restored.

The **thrashaction** parameter specifies the action the switch takes when it detects MAC address thrashing on a port. Thrashing occurs when one or more ports repeatedly learn the same MAC addresses, for example, as a result of a network loop.

Take care with the **thrashaction** parameter because misuse can impair your network operation.

Set the **thrashaction** parameter to:

- **none** to apply no thrash limiting to the port.
- **learndisable** to disable MAC address learning on the port, until the period specified with the **thrashtimeout** parameter has elapsed. The default is **learndisable**.
- **portdisable** or **linkdown** to disable the port until either the period specified by the **thrashtimeout** parameter has elapsed, or until the port is re-enabled by using the **enable switch port** command. If you specify **linkdown**, the link state is down; if you specify **portdisable**, the link state remains up.
- **vlandisable** to block all traffic on the VLAN where the address was learned, until either the period specified by **thrashtimeout** has elapsed, or until the port is re-enabled by using the **enable switch port vlan** command.

The **thrashtimeout** parameter specifies the time, in seconds, for which the switch employs the thrash action specified by the **thrashaction** parameter. The **thrashtimeout** cannot be set to **none** if **thrashaction=learndisable**. If **thrashtimeout=none**, and **thrashaction** is then changed to **learndisable**, then the switch automatically changes the **thrashtimeout** to 1 second.

If **none** is specified, the port is not automatically re-enabled, but can be re-enabled by using the **enable switch port** command for **thrashaction=portdisable** or **linkdisable**, and the **enable switch port vlan** command for **thrashaction=vlandisable**. The default is 1 second.

The **vlanstatustrap** parameter specifies whether the switch will send an SNMP trap whenever a port is enabled or disabled for a VLAN. A port can be disabled for a VLAN by using the **disable switch port** command, either when thrashing is detected on a port and the port's **thrashaction** is **vlandisable**, or when a storm is detected by QoS storm protection and the **stormaction** is **vlandisable**. If **on** is specified, a trap is sent. If **off** is specified, a trap is not sent. The default is **off**.

**Examples** To set all switch ports at the default maximum bandwidth, use the command:

```
set swi po=all egr=def
```

To set port 1 to transmit at a bandwidth of 500000 kbps, use the command:

```
set swi po=1 egr=500000
```

To set rate limiting of broadcast packets to 1000Kbytes per second for ports 10 and 12, use the command:

```
set swi po=10,12 bcl=1000
```

To disable rate limiting of broadcast packets for ports 10 and 12, and also disable multicast packet limiting as necessary, for this port only, use the command:

```
set swi po=10,12 bcl=0
```

To apply IGMP filter 1 to port 12, use the command:

```
set swi po=12 igmpfi=1
```

To limit the number of multicast groups that ports 12–23 can join to 50, use the command:

```
set swi po=12-23 igmpma=50
```

**Related Commands**

- [disable switch mclimiting](#)
- [disable switch port](#)
- [enable switch mclimiting](#)
- [enable switch port](#)
- [enable switch port vlan](#)
- [show switch port](#)

## set switch thrashlimit

---

**Syntax** SET SWITCh THRASHLimit=5..255

**Description** This command sets the maximum number of times a MAC address can move between ports, in one second. When the specified limit is reached, the **thrashaction** specified with the **set switch port** command is applied. The default **thrashlimit** is 10.

**Example** To set the switch thrash limit to 100 MAC movements per second, use the command:

```
set swi thrashl=100
```

**Related Commands**

- [set switch port](#)
- [show switch](#)



## set switch trunk

**Syntax for x900-24X** SET SWITCH TRunk=*trunk* [Speed={10M|100M|1000M|10G}]  
[THRASHAction={LEarndisable|LINKDown|NONE|Portdisable|  
VLANdisable}] [THRASHTimeout={None|1..86400}]

**Syntax for x900-48FE and AT-9900** SET SWITCH TRunk=*trunk* [Speed={10M|100M|1000M}]  
[THRASHAction={LEarndisable|LINKDown|NONE|Portdisable|  
VLANdisable}] [THRASHTimeout={None|1..86400}]

where *trunk* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The name is not case sensitive.

**Description** This command sets the speed for a specific trunk group on the switch. The switch supports static 802.3ad link aggregation, and port trunking is also called *link aggregation*.

The **trunk** parameter is the unique name of an existing trunk group.

The **speed** parameter lets you specify the speed of ports in a trunk group. Note the following:

The **speed** parameter lets you specify the speed of ports in a trunk group. Note the following:

- The **10M** and **100M** options are valid for 10/100 RJ-45 ports on x900-48FE switches.
- The **10M**, **100M**, and **1000M** options are valid for 10/100/1000 RJ-45 ports on x900-24X and AT-9900 switches.
- The **10M**, **100M**, and **1000M** options are valid for 10/100/1000 copper SFP ports on x900-24X and AT-9900 switches.
- The **100M** option is valid for 100Mb fibre SFP ports on AT-9900 switches.
- The **1000M** option is valid for copper SFP ports on x900-48FE switches.
- The **100M** option is valid for 100Mb SFP ports on x900-24XS switches.
- The **1000M** option is valid for 1Gb fibre SFP ports on all models.
- The **10G** option is valid for 10Gb fibre XFP ports only on x900-24X switches.

If the speed is not set, the trunk speed becomes the default speed of the master trunk port. The master trunk port is the lowest numbered port in the trunk group. The default is the highest speed the port is capable of.

When you add a port to a trunk group, the switch saves the port's current speed and duplex mode settings and sets the port to autonegotiate to the speed of the trunk group and full duplex mode. If the port does not support auto-negotiation, the switch sets the port to the fixed speed of the trunk group and full duplex mode.

The **thrashaction** parameter specifies the action the switch takes when it detects MAC address thrashing on a trunk. Thrashing occurs when one or more ports or trunks repeatedly learn the same MAC addresses, for example, as a result of a network loop. The switch applies the trunk's **thrashaction** to all ports in the trunk. Take care with the **thrashaction** parameter because misuse can impair your network operation.

Set the **thrashaction** parameter to:

- **none** to apply no thrash limiting on the trunk.
- **learndisable** to disable MAC address learning on all ports in the thrashing trunk, until the period specified with the **thrashtimeout** parameter has elapsed. The default is **learndisable**.
- **portdisable** or **linkdown** to disable all ports in the thrashing trunk until either the period specified by the **thrashtimeout** parameter has elapsed, or until the ports or subset of ports in the trunk are re-enabled by the **enable switch port** command. If you specify **linkdown**, the link state is down; if you specify **portdisable**, the link state remains up.
- **vlandisable** to block all traffic on the VLAN where the address was learned, on all ports in the thrashing trunk, until either the period specified by **thrashtimeout** has elapsed, or until the ports are re-enabled using the **enable switch port vlan** command. When **thrashaction=vlandisable**, there is only one timer per trunk, so if multiple VLANs have been disabled on a trunk, the timer starts when the last VLAN was disabled. When the timer expires, all VLANs are re-enabled on the trunk. When **thrashaction=vlandisable**, ingress filtering is automatically enabled on all ports in the trunk.

The **thrashtimeout** parameter specifies the time, in seconds, for which the switch employs the thrash action specified by the **thrashaction** parameter. The **thrashtimeout** cannot be set to **none** if **thrashaction=learndisable**. If **thrashtimeout=none**, and **thrashaction** is then changed to **learndisable**, then the switch automatically changes the **thrashtimeout** to 1 second.

If **none** is specified, the trunk is not automatically re-enabled, but individual ports can be re-enabled by using the **enable switch port** command for **thrashaction=portdisable** or **linkdisable**, and the **enable switch port vlan** command for **thrashaction=vlandisable**. The default is 1 second.

**Example** To set the speed of a trunk group called Trunk1 to 100Mbps, use the command:

```
set swi tr=trunk1 sp=100M
```

**Related Commands**

- [add switch trunk](#)
- [create switch trunk](#)
- [delete switch trunk](#)
- [destroy switch trunk](#)
- [enable switch port](#)
- [enable switch port vlan](#)
- [set switch trunk](#)
- [show switch port](#)
- [show switch trunk](#)

---

## set vlan port

---

**Syntax** SET VLAN={*vlan-name*|1..4094} Port={*port-list*|ALL}  
FRame={UNTAGged|TAGged}

where:

- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.
- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port.

**Description** This command changes the status of ports in a VLAN from tagged to untagged or from untagged to tagged. This command is effective on port-based VLANs only.

The **vlan** parameter specifies the name of the VLAN or the numerical VLAN identifier (VID) of the port-based VLAN. The name is not case sensitive, although the case is preserved for display purposes. The VLAN specified must exist.

The **port** parameter specifies the port(s) to be changed. The ports must belong to the VLAN specified. If the command succeeds on a subset of the ports specified but causes an error on the others, then the command as a whole fails and has no effect. If **all** is specified, then all ports in the VLAN are changed.

The **frame** parameter specifies whether a VLAN tag header is included in each frame transmitted on port-based VLANs. If **tagged** is specified, the port is then called a tagged port and a VLAN tag is added to frames prior to transmission.

If **untagged** is specified, the port is then called an untagged port and the frame is transmitted without a VLAN tag. A port can be untagged for one and only one of the VLANs to which it belongs, or for none of the VLANs to which it belongs. A port can have the **frame** parameter set to **tagged** for zero or more VLANs to which it belongs. It is not possible to add an untagged port to a VLAN if the port is already present in any other VLAN except the default VLAN.

If the port is an untagged member of the default VLAN's port association, adding it as untagged to another VLAN's port association deletes it from the default VLAN. The default setting is **untagged**.

**Example** To change the status of port 1 of the default VLAN from untagged to tagged, use the command:

```
set vlan=def po=1 fra=tag
```

**Related Commands**

- [add vlan subnet](#)
- [delete vlan port](#)
- [show vlan](#)

## show lacp

**Syntax** SHOW LACP

**Description** This command displays the state of LACP on the switch (Figure 7-16, Table 7-8).

Figure 7-16: Example output from the **show lacp** command

```
LACP Information
-----
Status ..... Enabled
Actor System Priority ..... 80-00
Actor System ..... 00-3e-0a-12-00-01
Address learn thrash action .... Learn Disable
Address learn thrash timeout .... 1 second
LACP Ports ..... 1-3,5,7,9-12
Active ..... 1-3,5
Passive ..... 7,9-12
```

Table 7-8: Parameters in output of the **show lacp** command

| Parameter                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                  |                                                |              |                                                                  |           |                                                                |              |                                                                               |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|------------------------------------------------|--------------|------------------------------------------------------------------|-----------|----------------------------------------------------------------|--------------|-------------------------------------------------------------------------------|
| Status                       | Whether LACP is enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                  |                                                |              |                                                                  |           |                                                                |              |                                                                               |
| Priority                     | User-configurable priority of the system. This parameter is concatenated with the Actor System parameter to generate the Actor System ID.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                  |                                                |              |                                                                  |           |                                                                |              |                                                                               |
| Actor System                 | MAC address of the local system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                  |                                                |              |                                                                  |           |                                                                |              |                                                                               |
| Address learn thrash action  | The <b>thrashaction</b> value that is applied to any trunks created by LACP. This specifies the action the switch takes when the address learn thrash limit is exceeded on the trunk. <table border="1"> <tr> <td>Disable Learning</td><td>Learning is disabled on all ports in the trunk</td></tr> <tr> <td>Disable Port</td><td>All ports in the trunk are disabled but the links will remain up</td></tr> <tr> <td>Link Down</td><td>All ports in the trunk are disabled and the links will go down</td></tr> <tr> <td>Disable Vlan</td><td>All ports in the trunk are disabled for the VLAN that thrashing occurring on.</td></tr> </table> | Disable Learning | Learning is disabled on all ports in the trunk | Disable Port | All ports in the trunk are disabled but the links will remain up | Link Down | All ports in the trunk are disabled and the links will go down | Disable Vlan | All ports in the trunk are disabled for the VLAN that thrashing occurring on. |
| Disable Learning             | Learning is disabled on all ports in the trunk                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                  |                                                |              |                                                                  |           |                                                                |              |                                                                               |
| Disable Port                 | All ports in the trunk are disabled but the links will remain up                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                  |                                                |              |                                                                  |           |                                                                |              |                                                                               |
| Link Down                    | All ports in the trunk are disabled and the links will go down                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                  |                                                |              |                                                                  |           |                                                                |              |                                                                               |
| Disable Vlan                 | All ports in the trunk are disabled for the VLAN that thrashing occurring on.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                  |                                                |              |                                                                  |           |                                                                |              |                                                                               |
| Address learn thrash timeout | The <b>thrashtimeout</b> value to apply to any trunks created by LACP. It specifies the time, in seconds, for which a trunk remains disabled after being disabled by thrashing protection.<br>If 'None' is shown, the trunk remains disabled until manually re-enabled.                                                                                                                                                                                                                                                                                                                                                                         |                  |                                                |              |                                                                  |           |                                                                |              |                                                                               |
| LACP Ports                   | A list of ports currently under LACP control.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                  |                                                |              |                                                                  |           |                                                                |              |                                                                               |
| Active                       | A list of ports currently in LACP Active mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                  |                                                |              |                                                                  |           |                                                                |              |                                                                               |
| Passive                      | A list of ports currently in LACP Passive mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                  |                                                |              |                                                                  |           |                                                                |              |                                                                               |

## show lacp port

**Syntax** `SHoW LACP Port [= {port-list | ALL}]`

where *port-list* is a port number, range (specified as *n-m*), or a comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.

**Description** This command displays LACP information about a specific switch port or all of them (Figure 7-17, Table 7-9).

Figure 7-17: Example output from the **show lacp port** command

```

LACP Port Information
-----
Actor Port ..... 1
Trunk Group ..... lacp1
Selected ..... Selected
Port Priority .....8000
LACP Port Number ..... 0001
Port Key ... .....6
Admin Key ..... 12
Mode ..... Active
Periodic..... Fast
Individual ..... No
Synchronised .....Yes
Collecting .....Yes
Distributing ..... Yes
Defaulted ..... No
Expired ..... No
Actor Churn.....No
Partner Churn.....No
Partner Information
Partner System Priority ..... 8000
Partner System ..... 00-3e-0a-12-00-01
Port Key .....4
Port Priority ..... 500
Port Number ..... 0002
Mode ..... Active
Periodic..... Fast
Individual ..... No
Synchronised .....Yes
Collecting .....Yes
Distributing ..... Yes
Defaulted ..... No
Expired ..... No
-----

```

Table 7-9: Parameters in output of the **show lacp port** command

| Parameter        | Meaning                                                                                                                                                                                                                                                                                                                                                                             |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Port             | Number of the port.                                                                                                                                                                                                                                                                                                                                                                 |
| Trunk Group      | Name of trunk group to which the port belongs. It is a name that LACP has automatically assigned to an aggregated link. You cannot manually create a trunk starting with the letters LACP. If LACP created, then the name has the prefix LACP followed by a numeric, such as LACP72. This number is the same as the new interface index shown by the <b>show interface</b> command. |
| Priority         | User-configurable priority assigned to the port.                                                                                                                                                                                                                                                                                                                                    |
| LACP Port Number | LACP encoded port number.                                                                                                                                                                                                                                                                                                                                                           |
| Port Key         | Key that LACP has assigned to the port.                                                                                                                                                                                                                                                                                                                                             |
| Admin Key        | User-configurable key assigned to the port.                                                                                                                                                                                                                                                                                                                                         |
| Mode             | The participation mode. If active, the port sends LACPDU packets regardless of the partner port's participation. If passive, the port sends LACPDU packets after receiving one from its partner port.                                                                                                                                                                               |
| Periodic         | User-configurable time period between transmission of periodic LACPDU packets; either fast (1 second) or slow (30 seconds).                                                                                                                                                                                                                                                         |
| Individual       | User-configurable setting that determines whether the port is an individual. If no, the port may be aggregated; if yes, it is not aggregated.                                                                                                                                                                                                                                       |

Table 7-9: Parameters in output of the **show lacp port** command (cont)

| Parameter               | Meaning                                                                                                                                                                                                                     |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synchronised            | If yes, the port is considered to be in a synchronised state—the port has been correctly associated with an aggregator.                                                                                                     |
| Collecting              | Whether this port has been enabled to receive packets.                                                                                                                                                                      |
| Distributing            | Whether this port has been enabled to transmit packets.                                                                                                                                                                     |
| Defaulted               | Whether this system is using defaults for the partner information. If no, the values have been received from the partner via a LACPDU.                                                                                      |
| Expired                 | The port has not received a frame from its partner for 3 times the periodic time (3 or 90 seconds).                                                                                                                         |
| Actor Churn             | Whether churning of the actor port has been detected.                                                                                                                                                                       |
| Partner Churn           | Whether churning of the partner port has been detected.                                                                                                                                                                     |
| Partner Information     | Information that has been received about the partner port. The partner port is the port on the connected device.                                                                                                            |
| Partner System Priority | Partner's system priority.                                                                                                                                                                                                  |
| Partner System          | Partner's system identifier.                                                                                                                                                                                                |
| Port Key                | Partner port's key.                                                                                                                                                                                                         |
| Port Priority           | Partner port's key priority.                                                                                                                                                                                                |
| Port Number             | Partner port's port number.                                                                                                                                                                                                 |
| Mode                    | Whether the mode is active or passive. If active, the partner port sends LACPDU packets regardless of this port's participation. If passive, the partner port sends LACPDU packets only after receiving one from this port. |
| Periodic                | The setting of the partner port for the time period between transmission of periodic LACPDU packets; either fast (1 second) or slow (30 seconds).                                                                           |
| Individual              | The setting of the partner port determining whether the port is an individual. If no, the partner port is not an individual and may be aggregated; if yes, it cannot be aggregated.                                         |
| Synchronised            | If yes, the partner system considers the partner port to be in a synchronised port—the port has been correctly associated with an aggregator; otherwise, no.                                                                |
| Collecting              | Whether the partner port has been enabled for receiving packets.                                                                                                                                                            |
| Distributing            | Whether the partner port has been enabled for transmitting packets.                                                                                                                                                         |
| Defaulted               | Whether the partner system is using the defaults for this port's information. If no, the values have been received from this system via a LACPDU. If yes, the defaults are still in use.                                    |
| Expired                 | When the partner port has not received a frame for 3 times the periodic time (3 or 90 seconds).                                                                                                                             |

**Examples** To show the LACP port information for all ports, use the command:

```
sh lacp po
```

**Related Commands**

- [add lacp port](#)
- [delete lacp port](#)
- [set lacp port](#)
- [show lacp](#)

## show lacp port counter

**Syntax** SHow LACP Port[={*port-list*|ALL}] COUnter

where *port-list* is a port number, range (specified as *n-m*), or a comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.

**Description** This command displays LACP counters for all switch ports or specific ones (Figure 7-18, Table 7-10).

Figure 7-18: Example output from the **show lacp port counter** command

```
LACP Port Counters
-----
Port 1
  Received:                                Transmitted:
  LACP Pkts..... 0                      LACP Pkts ..... 0
  Invalid LACP Pkts..... 0
-----
```

Table 7-10: Parameters in output of the **show lacp port counter** command

| Parameter          | Meaning                                                                                                                                                                                                                                                                            |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Received</b>    | Counters for received LACP packets                                                                                                                                                                                                                                                 |
| LACP Pkts          | Number of valid LACPDU frames received.                                                                                                                                                                                                                                            |
| Invalid LACP Pkts  | Number of invalid LACP packets received. This includes those with an invalid type/length field, subtype field, actor information length field, partner information length field, collector information length field, terminator information length field, or invalid frame length. |
| <b>Transmitted</b> | Counters for transmitted LACP packets                                                                                                                                                                                                                                              |
| LACP Pkts          | Number of LACPDU frames transmitted.                                                                                                                                                                                                                                               |

**Examples** To show the LACP port counters for all ports, use the command:

```
sh lacp po cou
```

**Related Commands** [reset lacp port counter](#)  
[show lacp](#)  
[show lacp port](#)

## show lacp trunk

---

**Syntax** SHow LACP TRunk

**Description** This command displays information about the dynamically configured trunks currently configured for LACP ([Figure 7-19](#)).

Figure 7-19: Example output from the **show lacp trunk** command

```
LACP Dynamic Trunk Group Information
-----

Trunk group name ..... lacp53:
  Speed ..... 100 Mbps
  Ports in Trunk ..... 10,15
  LAG ID:
  [ (8000,00-00-cd-03-00-79,0005,00,0000) , (8000,00-00-cd-08-76-60,0002,00,0000) ]
-----
```

**Related Commands** [show lacp trunk](#)  
[show lacp](#)

## show switch

---

**Syntax** SHow SWITch

**Description** This command displays configuration information about the switch functions.

- For x900-48FE switches, see [Figure 7-20 on page 7-121](#).
- For AT-9900 and x900-24X switches, see [Figure 7-21 on page 7-122](#).

For a description of parameters, see [Table 7-11 on page 7-123](#).



Figure 7-20: Example output from the **show switch** command for x900-48FE switches

```

Switch Configuration
-----
Switch Address ..... 00-00-cd-12-78-03
Learning ..... ON
Ageing Timer ..... ON
Number of Fixed Ports ..... 48
Number of Uplink Ports ..... 4
Mirroring ..... ENABLED
Mirror port ..... 34
Ports mirroring on Rx ..... 8,11
Ports mirroring on Tx ..... None
Ports mirroring on Both .... 6
Nested TPID ..... 0x8100
Number of WAN Interfaces ... 0
Name of Interface(s) ..... -
Ageingtime ..... 300
DLF rate limit ..... -
STP Forwarding ..... Disabled
UpTime ..... 00:27:09
Hashingfield ..... L2 L3
IP route:
    Learn delay ..... 4 ms
        queue size ..... 0
        queue limit ..... 1000000
        percent in use .... 0
        high water mark ... 0
        queue maximum ..... 1500000
        queue default ..... 1000000
    Updating hardware(status) 0 (Pending)
-----
Traffic Control Unit, hardware resource usage:
Total system rule space ..... 1024
Total number of rules used .... 35
Total rule space usage ..... 64
Number of rules per application:
    MLD Snooping ..... 8
    Accel. Card(IPv6) ..... 1
    Switch HwFilter ..... 20
    QOS ..... 6
Total number of actions ..... 1024
Number of actions used ..... 37
Device Resource, device #0:
    Number of rules used ..... 34
    Rule space usage ..... 56
    Number of rules per application:
        MLD Snooping ..... 8
        Switch HwFilter ..... 20
        QOS ..... 6
    Device rule space limit ..... 1024
Profile Usage:
    Profile #1:
        IPv4 bytes used ..... 16 of 16
        Other-Eth bytes used .... 5 of 16
Device Resource, device #1:
    Number of rules used ..... 1
    Rule space usage ..... 8
    Number of rules per application:
        Accel. Card(IPv6) ..... 1
    Device rule space limit ..... 1024
Profile Usage:
    Profile #1:
        IPv4 bytes used ..... 0 of 16
        Other-Eth bytes used .... 6 of 16

```

Figure 7-21: Example output from the **show switch** command for AT-9900 and x900-24X switches

```

Switch Configuration
-----
Switch Address ..... 00-00-cd-10-00-9b
Learning ..... ON
Ageing Timer ..... ON
Jumbo Frames ..... OFF
Number of Fixed Ports ..... 24
Number of Uplink Ports ..... 0
Mirroring ..... DISABLED
Mirror port ..... None
Ports mirroring on Rx ..... None
Ports mirroring on Tx ..... None
Ports mirroring on Both .... None
Nested TPID ..... 0x8100
Number of WAN Interfaces ... 0
Name of Interface(s) ..... -
Ageingtime ..... 300
DLF rate limit ..... -
STP Forwarding ..... Disabled
UpTime ..... 00:41:54
Hashingfield ..... L2 L3
IP route:
  Learn delay ..... 4 ms
  queue size ..... 0
  queue limit ..... 1000000
  percent in use .... 0
  high water mark ... 0
  queue maximum ..... 1500000
  queue default ..... 1000000
  Updating hardware(status) 0 (Pending)
-----
Traffic Control Unit, hardware resource usage:
Total system rule space ..... 2048
Total number of rules used .... 4
Total rule space usage ..... 16
Number of rules per application:
  MLD Snooping ..... 4
Total number of actions ..... 2048
Number of actions used ..... 13959168
Device Resource, device #0:
  Number of rules used ..... 2
  Rule space usage ..... 8
  Number of rules per application:
    MLD Snooping ..... 2
  Device rule space limit ..... 2048
Profile Usage:
  Profile #1:
    IPv4 bytes used ..... 2 of 16
    Other-Eth bytes used .... 7 of 16
Device Resource, device #1:
  Number of rules used ..... 2
  Rule space usage ..... 8
  Number of rules per application:
    MLD Snooping ..... 2
  Device rule space limit ..... 2048
Profile Usage:
  Profile #1:
    IPv4 bytes used ..... 2 of 16
    Other-Eth bytes used .... 7 of 16

```

Table 7-11: Parameters in output of the **show switch** command

| Parameter                | Meaning                                                                                                                                                                                               |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Switch Address           | MAC address of the switch, from which the bridge identifier used in the Spanning Tree algorithm is derived. This address is used as the source MAC address in pause frames transmitted by the switch. |
| EnhancedMode operation   | Whether the switch's memory has been rearranged to store either QoS traffic class counters (QOSCOUNTERS), or a greater number of routes than usual (NEXTHOP), or is left in its default state (NONE). |
| Learning                 | Whether the switch's dynamic learning and updating of the forwarding database is enabled.                                                                                                             |
| Ageing Timer             | Whether the ageing timer is enabled.                                                                                                                                                                  |
| Jumbo Frames             | Whether the jumbo frames feature is enabled on AT-9900 switches.                                                                                                                                      |
| Number of Fixed Ports    | Number of fixed Ethernet switch ports.                                                                                                                                                                |
| Number of Uplink Ports   | Number of Ethernet uplink ports.                                                                                                                                                                      |
| Mirroring                | Whether traffic mirroring is enabled.                                                                                                                                                                 |
| Mirror port              | Switch port where mirror traffic is sent.                                                                                                                                                             |
| Ports mirroring on Rx    | Ports that are set to send all the traffic they receive to the mirror port.                                                                                                                           |
| Ports mirroring on Tx    | Ports that are set to send all the traffic they transmit to the mirror port.                                                                                                                          |
| Ports mirroring on Both  | Ports that are set to send all the traffic they both receive and transmit to the mirror port.                                                                                                         |
| Nested TPID              | Ethernet type of a tag added by a nested VLAN. This is set by default to 0x8100 when a nested VLAN is created                                                                                         |
| Number of WAN Interfaces | Total number of installed WAN interfaces.                                                                                                                                                             |
| Name of Interface(s)     | Name of the installed WAN interface(s).                                                                                                                                                               |
| Ageingtime               | Seconds after which a dynamic entry is removed from the forwarding database.                                                                                                                          |
| DLF rate limit           | Limit of the rate of reception of destination lookup failure packets for the switch, in Kbytes per second.                                                                                            |
| STP Forwarding           | Whether STP forwarding is enabled.                                                                                                                                                                    |
| Uptime                   | Time in hours:minutes:seconds since the SWITCH was last powered up, rebooted, or restarted. This is the same as the value of the MIB object sysUpTime.                                                |
| Hashingfield             | Packet fields used in the Link Aggregate hash function.                                                                                                                                               |
| <b>IP route</b>          | <b>The following fields display information about the process of copying routes from the software routing table to the hardware routing table.</b>                                                    |
| Learn delay              | Number of milliseconds that the switch waits after the last IP route is inserted before it starts to update the hardware routing system.                                                              |
| Queue size               | The number of entries currently in the hardware route update queue.                                                                                                                                   |
| Queue limit              | The maximum number of entries that the queue can hold.                                                                                                                                                |
| Percent in use           | The percentage of the queue limit that is currently used.                                                                                                                                             |

Table 7-11: Parameters in output of the **show switch** command (cont)

| Parameter                         | Meaning                                                                                                                                                                                                                                          |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| High water mark                   | The highest number of messages that have been seen on the queue since the switch last started up.                                                                                                                                                |
| Queue maximum                     | The maximum value to which you can set the queue size. This depends on the amount of memory installed on the switch.                                                                                                                             |
| Queue default                     | The default maximum number of entries in the queue. This depends on the amount of memory installed on the switch.                                                                                                                                |
| Updating hardware (status)        | The number of entries that the software has queued for writing into the hardware table, followed by the status. Status is Pending if the hardware is not currently processing queued routes and Active if it is currently processing the routes. |
| <b>Traffic Control Unit...</b>    | <b>The following fields display information about the hardware resources used by the Traffic Control Unit.</b>                                                                                                                                   |
| Total system rule space           | Total number of classifiers/filter rules available in the system.                                                                                                                                                                                |
| Total number of rules used        | Number of classifiers currently being used.                                                                                                                                                                                                      |
| Total rule space usage            | Number of rules reserved, accounting for block size of 8.                                                                                                                                                                                        |
| Number of rules per application   | A list of applications, and the number of classifiers used by each application.                                                                                                                                                                  |
| Total number of actions           | Total number of actions able to be used by hardware filters or QoS.                                                                                                                                                                              |
| Number of actions used            | Number of actions currently being used.                                                                                                                                                                                                          |
| <b>Device Resource, device #n</b> | <b>The following fields display information about the hardware resources used by a specific device, and are repeated for each device:</b><br><b>#0: base unit</b><br><b>#1: accelerator card</b>                                                 |
| Number of rules used              | Total number of classifiers being used by this device.                                                                                                                                                                                           |
| Rule space usage                  | Number of rules reserved, accounting for block size of 8.                                                                                                                                                                                        |
| Number of rules per application   | A list of applications, and the number of classifiers used by each application on this device.                                                                                                                                                   |
| Device rule space limit           | Total number of classifiers available for use by this device.                                                                                                                                                                                    |
| Profile #1                        | Profile used to match on packets on this device.                                                                                                                                                                                                 |
| IPv4 bytes used                   | Number of bytes of the profile being used to match IPv4.                                                                                                                                                                                         |
| IPv6 bytes used                   | Number of bytes of the profile being used to match IPv6.                                                                                                                                                                                         |
| Other-Eth bytes used              | Number of bytes of the profile being used to match non-IP Ethernet frames.                                                                                                                                                                       |

**Example** To display the configuration of the switch, use the command:

```
sh swi
```

**Related Commands**

- [create vlan](#)
- [enable switch hash](#)
- [reset switch](#)
- [set switch ageingtimer](#)
- [set switch mirror](#)
- [set switch port](#)

## show switch accelerator

**Syntax** SHow SWItch ACCELerator

**Description** This command displays information about the IPv6 accelerator card for AT-8948 switches (Figure 7-22, Table 7-12).

Figure 7-22: Example output from the **show switch accelerator** command

```
Switch Accelerator Configuration
-----
Hardware Type ..... AT-ACC01
Mode ..... IPv6 Acceleration
Status ..... IPv6 active
Search memory size ..... 128 Mb
Counter memory size ..... 2 Mb
MAC address ..... 00-00-CD-80-00-66
-----
```

Table 7-12: Parameters in output of the **show switch accelerator** command

| Parameter           | Meaning                                                                                                        |
|---------------------|----------------------------------------------------------------------------------------------------------------|
| Hardware Type       | Model name of the accelerator card.                                                                            |
| Mode                | Whether the card is functioning in IPv6 Acceleration, loopback, or disabled mode.                              |
| Status              | Whether the IPv6 accelerator card is active, missing, disabled, or has an error.                               |
| Search memory size  | Size of the card's Search memory space. The Search memory holds data such as multicast tables and route trees. |
| Counter memory size | Size of the card's Counter memory space. The Counter memory holds diagnostic counters.                         |
| MAC address         | MAC address that is used for IPv6 routing.                                                                     |

**Examples** To display information about the IPv6 accelerator card, use the command:

```
sh swi accel
```

**Related Commands** [delete switch accelerator hwfilter](#)  
[disable switch accelerator](#)

## show switch accelerator counter

**Syntax** Show Switch ACCELerator COUnTer [= {ALL | DEFault | FAB | MAC | MIB | MISc}]

**Description** This command displays counters stored in the IPv6 accelerator card for AT-8948 switches (Figure 7-23, Table 7-13 on page 7-127).

The **counter** parameter specifies which block of counters to display. If **all** is specified, all counters are displayed. If **default** is specified, the MAC and MIB counters are displayed. If **fab** is specified, the internal IPv6 accelerator card fabric interface counters are displayed. If **mac** is specified, the internal IPv6 accelerator card MAC counters are displayed. If **mib** is specified, mode-specific SNMP counters, cumulative across all interfaces, are displayed. If **MISC** is specified, miscellaneous IPv6 accelerator card status counters are listed. The default is **default**.

For more information about accelerated IPv6 traffic, see [“IPv6 Acceleration” on page 23-21 of Chapter 23, Internet Protocol version 6 \(IPv6\)](#).

Figure 7-23: Example output from the **show switch accelerator counter** command

| Switch Accelerator Counters                                  |       |                |      |
|--------------------------------------------------------------|-------|----------------|------|
| -----                                                        |       |                |      |
| Accelerator Ethernet MAC counters:                           |       |                |      |
| Combined receive/transmit packets by size (octets) counters: |       |                |      |
| 64                                                           | 2     | 512 - 1023     | 14   |
| 65 - 127                                                     | 220   | 1024 - 1518    | 15   |
| 128 - 255                                                    | 6     | 1519 - 1522    | 0    |
| 256 - 511                                                    | 10    |                |      |
| General Counters:                                            |       |                |      |
| Receive                                                      |       | Transmit       |      |
| Octets                                                       | 41652 | Octets         | 7004 |
| Pkts                                                         | 164   | Pkts           | 103  |
| FCSErrors                                                    | 2     | FCSErrors      | 0    |
| MulticastPkts                                                | 8     | MulticastPkts  | 0    |
| BroadcastPkts                                                | 8     | BroadcastPkts  | 0    |
| PauseCtrlFrms                                                | 0     | PauseCtrlFrms  | 0    |
| OversizePkts                                                 | 0     | OversizePkts   | 0    |
| Fragments                                                    | 0     | Fragments      | 0    |
| Jabbers                                                      | 0     | Jabbers        | 0    |
| UndersizePkts                                                | 0     | UndersizePkts  | 0    |
| Control                                                      | 0     | Control        | 0    |
| LengthError                                                  | 0     | LengthError    | 0    |
| CodeError                                                    | 0     |                |      |
| UnknownOffset                                                | 0     |                |      |
| Accelerator IPv6 MIB counters:                               |       |                |      |
| InReceives                                                   | 154   | OutForwDatagrm | 43   |
| InNoRoutes                                                   | 60    |                |      |
| InDiscards                                                   | 0     | OutDiscards    | 0    |
| InAddrErrors                                                 | 0     |                |      |
| InTruncatedPkts                                              | 43    |                |      |
| InMcastPkts                                                  | 8     | OutMcastPkts   | 0    |
| InDelivers                                                   | 0     |                |      |
| InHdrErrors                                                  | 0     |                |      |
| InTooBigErrors                                               | 0     |                |      |

Table 7-13: Parameters in output from the **show switch accelerator counter** command

| Parameter                                                   | Description                                                                   |
|-------------------------------------------------------------|-------------------------------------------------------------------------------|
| <b>Accelerator Ethernet MAC counters</b>                    |                                                                               |
| Combined receive/transmit packets by size (octets) counters | Number of packets in each size range received and transmitted.                |
| 64                                                          | Number of 64 octet packets received and transmitted.                          |
| 65 - 127                                                    | Number of 65 - 127 octet packets received and transmitted.                    |
| 128 - 255                                                   | Number of 128 - 255 octet packets received and transmitted.                   |
| 256 - 511                                                   | Number of 256 - 511 octet packets received and transmitted.                   |
| 512 - 1023                                                  | Number of 512 - 1023 octet packets received and transmitted.                  |
| 1024 - 1518                                                 | Number of 1024 - 1518 octet packets received and transmitted.                 |
| 1519 - 1522                                                 | Number of 1519 - 1522 octet frames received and transmitted.                  |
| <b>General Counters</b>                                     |                                                                               |
| <b>Receive</b>                                              | Counters for traffic received.                                                |
| Octets                                                      | Number of octets received.                                                    |
| Pkts                                                        | Number of packets received.                                                   |
| FCSErrors                                                   | Number of frames received that contain a Frame Check Sequence error.          |
| MulticastPkts                                               | Number of multicast packets received.                                         |
| BroadcastPkts                                               | Number of broadcast packets received.                                         |
| PauseMACCtlFrms                                             | Number of valid PAUSE MAC Control frames received.                            |
| OversizePkts                                                | Number of oversize packets received.                                          |
| Fragments                                                   | Number of fragments received.                                                 |
| Jabbers                                                     | Number of jabber frames received.                                             |
| UndersizePkts                                               | Number of packets received that are smaller than 64 bytes.                    |
| Control                                                     | Number of control frames received.                                            |
| LengthError                                                 | Number of packets received with length errors.                                |
| CodeError                                                   | Number of frames received that contain error symbols or an invalid structure. |
| UnknownOffset                                               | Number of control frames (type 88-08) received that have an unknown opcode.   |
| <b>Transmit</b>                                             | Counters for traffic transmitted                                              |
| Octets                                                      | Number of octets transmitted.                                                 |
| Pkts                                                        | Number of packets transmitted.                                                |
| FCSErrors                                                   | Number of frames transmitted that contain a Frame Check Sequence error.       |
| MulticastPkts                                               | Number of multicast packets transmitted.                                      |
| BroadcastPkts                                               | Number of broadcast packets transmitted.                                      |

Table 7-13: Parameters in output from the **show switch accelerator counter** command (cont)

| Parameter                            | Description                                                                                              |
|--------------------------------------|----------------------------------------------------------------------------------------------------------|
| PauseCtrlFrms                        | Number of Pause control frames transmitted.                                                              |
| OversizePkts                         | Number of oversize packets transmitted.                                                                  |
| Fragments                            | Number of fragments transmitted.                                                                         |
| Jabbers                              | Number of jabber frames transmitted.                                                                     |
| UndersizePkts                        | Number of packets transmitted that were smaller than 64 bytes.                                           |
| Control                              | Number of control frames transmitted.                                                                    |
| LengthError                          | Number of packets transmitted with length errors.                                                        |
| <b>Accelerator IPv6 MIB counters</b> |                                                                                                          |
| InReceives                           | Number of packets received on the interface.                                                             |
| InNoRoutes                           | Number of input packets discarded because no route could be found to transmit them to their destination. |
| InDiscards                           | Number of incoming packets that were discarded for reasons other than errors in the packet.              |
| InAddrErrors                         | Number of packets received with invalid addresses.                                                       |
| InTruncatedPkts                      | Number of packets received that were truncated.                                                          |
| InMcastPkts                          | Number of multicast packets received.                                                                    |
| InDelivers                           | Number of incoming packets that were successfully delivered to a higher layer protocol.                  |
| InHdrErrors                          | Number of packets received with invalid headers.                                                         |
| InTooBigErrors                       | Number of packets received that were discarded because they were too big.                                |
| OutForwDatagrm                       | Number of packets that have been forwarded.                                                              |
| OutDiscards                          | Number of outgoing packets that were discarded for reasons other than errors in the packet.              |
| OutMcastPkts                         | Number of multicast packets sent.                                                                        |

**Examples** To display all IPv6 accelerator card counters, use the command:

```
sh swi accel cou=all
```

**Related Commands** [disable switch accelerator](#)  
[enable switch accelerator](#)



## show switch accelerator debug

**Syntax** `SHoW SWITCh ACCELeRator DEBUg`

**Description** This command shows the current debugging modes enabled on the IPv6 accelerator card on AT-8948 switches (Figure 7-24, Table 7-14).

For more information about accelerated IPv6 traffic, see “IPv6 Acceleration” on page 23-21 of Chapter 23, Internet Protocol version 6 (IPv6).

Figure 7-24: Example output from the **show switch accelerator debug** command

```
Enabled Switch Accelerator Debugging
-----
Debug output device ..... 16
Debug timeout ..... 0 seconds
Persistent debug flags ..... 0
Current debug flags ..... 80530080
Debug flags active:
  USEFUL Useful flags ..... 80530080
  ACCELERATOR Accelerator ..... 80000080
  INT_ERR Driver error ..... 10000
  INT_TRC Driver trace ..... 20000
  INT_CFG Driver configuration ..... 100000
  INT_API Driver API ..... 400000
-----
```

Table 7-14: Parameters in output of the **show switch accelerator debug** command

| Parameter              | Description                                                                                                                  |
|------------------------|------------------------------------------------------------------------------------------------------------------------------|
| Debug output device    | Internal identifier of the port where debug output is sent.                                                                  |
| Debug timeout          | Time in seconds after which debugging automatically ceases. A zero value indicates no timeout.                               |
| Persistent debug flags | Hexadecimal debug mask that is retained across restarts. A zero indicates that no debug flags are retained.                  |
| Current debug flags    | Hexadecimal debug mask that is currently enabled. The debug mask is the result of an OR operation on all active debug flags. |
| Debug flags active     | Debug modes that are currently active and their hexadecimal flags.                                                           |

**Examples** To display the current debugging modes enabled on the IPv6 accelerator card, use the command:

```
sh swi accel deb
```

**Related Commands** [disable switch accelerator debug](#)  
[enable switch accelerator debug](#)

## show switch accelerator hwfilter

**Syntax** `SHoW SWItch ACCELerator HWFilter[={filter-id|ALL}]`

where *filter-id* is a decimal number from 1 to 2000

**Description** This command displays configuration information for the hardware-based packet filters on the IPv6 accelerator card (Figure 7-25, Table 7-15). This command is valid for AT-8948 switches only.

Figure 7-25: Example output from the **show switch accelerator hwfilter** command

```

Accelerator Hardware-based Packet Filters
-----
Rule position..... 1
Rule Id..... 1
Action..... DISCARD

Rule position..... 2
Rule Id..... 2
Action..... MARK
      New priority ..... 3

Rule position..... 3
Rule Id..... 3
Action..... MARK
      New DSCP ..... 47
-----

```

Table 7-15: Parameters in output of the **show switch accelerator hwfilter** command

| Parameter     | Description                                                                                                                                |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Filter        | Identifier for the hardware-based packet filter.                                                                                           |
| Rule Position | Relative order that the classifier was added to the hardware-based packet filter. The order numbering starts at zero                       |
| Rule Id       | Identifier for the classifier.                                                                                                             |
| Action        | Action associated with the classifier entry for the hardware-based packet filter. Options include discard, send to CPU, mark, and forward. |
| New priority  | The 802.1q priority to assign to the packet when <i>action</i> is <i>mark</i> .                                                            |
| New DSCP      | The IPv6 DSCP value to assign to the packet when <i>action</i> is <i>mark</i> .                                                            |

**Examples** To display configuration information for the hardware based packet filters on the IPv6 accelerator card, use the command:

```
sh swi accel hwf
```

**Related Commands**

- [add switch accelerator hwfilter](#)
- [delete switch accelerator hwfilter](#)
- [show switch accelerator](#)
- [show switch accelerator counter](#)
- [show switch accelerator hwfilter](#)

## show switch counter

**Syntax** SHow SWItch COUnTer

**Description** This command displays information about the forwarding counters associated with the switch. The syntax and output for this command depends on the switching chipset in the switch ([Figure 7-26](#), [Table 7-16 on page 7-132](#)).

Figure 7-26: Example output from the **show switch counter** command

```
Switch Counters
-----
Switch instance:      0

Packet DMA counters:

Receive:              Transmit:
Packets                6      Packets                6
Discards               0      Discards               0
TooFewBuffers          0      Aborts                 0
DescriptorsExhausteds  0      DescriptorAreaFilleds  0
QueueLength            0      QueueLength            0

PCI bus counters:
ParityErrors           0      ErrorChannel           0
FatalErrors            0      ErrorResets            0

General counters:
Resets                 0
-----
Switch instance:      1

Packet DMA counters:

Receive:              Transmit:
Packets                2      Packets                2
Discards               0      Discards               0
TooFewBuffers          0      Aborts                 0
DescriptorsExhausteds  0      DescriptorAreaFilleds  0
QueueLength            0      QueueLength            0

PCI bus counters:
ParityErrors           0      ErrorChannel           0
FatalErrors            0      ErrorResets            0

General counters:
Resets                 0
-----
```

Table 7-16: Parameters in output of the **show switch counter** command

| Parameter                  | Meaning                                                                                                                                                                                                                                           |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Packet DMA counters</b> |                                                                                                                                                                                                                                                   |
| <b>Receive</b>             | Counters for packets received.                                                                                                                                                                                                                    |
| Packets                    | Number of packets received by the CPU from the switch chip.                                                                                                                                                                                       |
| Discards                   | Number of packets received from the switch chip that were discarded because either the receive queue was greater than 4096, or because the free buffers in the switch were below BufferLevel3, or because there were no data bytes in the packet. |
| TooFewBuffers              | Number of packets received from the switch chip that were discarded because the free buffers in the switch were below BufferLevel3.                                                                                                               |
| DescriptorsExhausteds      | Number of times the switch chip reported that it could not transfer a packet by DMA to a switch buffer because there were no more receive buffer descriptors.                                                                                     |
| QueueLength                | Number of packets received from the switch chip waiting to be processed by the CPU.                                                                                                                                                               |
| <b>Transmit</b>            | Counters for packets transmitted.                                                                                                                                                                                                                 |
| Packets                    | Number of packets transferred from the CPU to the switch chip.                                                                                                                                                                                    |
| Discards                   | Number of packets waiting for transmission that were discarded when the DMA process was reset due to an error.                                                                                                                                    |
| Aborts                     | Number of times transmission of a packet was aborted due to it taking an excessive length of time for the transmission to complete, perhaps due to a port being in a blocked state or due to a busy PCI bus.                                      |
| DescriptorAreaFilleds      | Number of times the transmit descriptor area filled due to a high rate of transfer of packets from the CPU to the switch chip or high PCI bus utilisation causing the DMA to proceed slowly.                                                      |
| QueueLength                | Number of packets currently queued for transmission, or that have been transmitted and are waiting to be purged from the transmit queue.                                                                                                          |
| <b>PCI bus counters</b>    |                                                                                                                                                                                                                                                   |
| ParityErrors               | Number of times the switch chip reported a parity error for a transaction on the PCI bus.                                                                                                                                                         |
| FatalErrors                | Number of times the switch chip reported a fatal error for a transaction on the PCI bus.                                                                                                                                                          |
| ErrorChannel               | The DMA channel for making the transaction for which the error occurred.                                                                                                                                                                          |
| ErrorResets                | Not currently supported.                                                                                                                                                                                                                          |
| <b>General counters</b>    |                                                                                                                                                                                                                                                   |
| Resets                     | The number of times the receive and transmit DMA channels have been reset due to the occurrence of an error.                                                                                                                                      |

**Examples** To display the switching counters, use the command:

```
sh swi cou
```

**Related Commands** [set switch port](#)  
[show switch](#)

## show switch debug

**Syntax** `SHoW SWItch DEBUg`

**Description** This command displays debugging information for the switch (Figure 7-27, Table 7-17).

Figure 7-27: Example output from the **show switch debug** command

|                            |        |         |
|----------------------------|--------|---------|
| Manager > sh swi debug     |        |         |
| Enabled Switch Debug Modes | Output | Timeout |
| -----                      |        |         |
| DMA                        | 16     | None    |

Table 7-17: Parameters in output of the **show switch debug** command

| Parameter                  | Meaning                                                                                                |
|----------------------------|--------------------------------------------------------------------------------------------------------|
| Enabled Switch Debug Modes | Whether the debugging option for the switch is DMA, QOS, PHY, or none.                                 |
| Output                     | Output device for the switch and shown when a debug mode is enabled.                                   |
| Timeout                    | Time in seconds that debugging options for the switch are enabled. Shown when a debug mode is enabled. |

**Example** To display debugging information for the switch, use the command:

```
sh swi deb
```

**Related Commands** [disable switch debug](#)  
[enable switch debug](#)  
[disable debug active](#) in Chapter 4, Configuring and Monitoring the System  
[show debug active](#) in Chapter 4, Configuring and Monitoring the System

## show switch fdb

**Syntax** Show Switch FDB [Address=*macadd*] [Port={*port-list*|ALL}]  
[Status={STATIC|DYNAMIC}] [VLAN={*vlan-name*|1..4094}]

where:

- *macadd* is an Ethernet six-octet MAC address, expressed as six pairs of hexadecimal digits delimited by hyphens.
- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.
- *vlan-name* is a unique name for the VLAN from 1 to 32 characters long. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.

**Description** This command displays the contents of the forwarding database (FDB) (Figure 7-28, Table 7-18). This command requires a user with Security Officer privilege when the switch is in security mode.

The **address** parameter specifies the MAC address of the device for which the contents of the forwarding database are to be displayed.

The **port** parameter specifies entries to be displayed from the forwarding database that were learned from the specified port.

The **status** parameter specifies whether to display static entries or dynamically-learned filter entries.

The **vlan** parameter specifies the VLAN identifier of the VLAN for which the contents of the forwarding database are to be displayed.

Figure 7-28: Example output from the **show switch fdb** command

| Switch Forwarding Database (software) |                   |           |         |         |
|---------------------------------------|-------------------|-----------|---------|---------|
| Total number of entries = 2           |                   |           |         |         |
| VLAN                                  | MAC Address       | Port/Vidx | Status  | daRoute |
| 1                                     | 00-00-cd-10-00-44 | 1         | dynamic | 0       |
| 1                                     | 00-00-cd-10-00-91 | CPU       | static  | 1       |

Table 7-18: Parameters in output of the **show switch fdb** command

| Parameter   | Meaning                                                                                                |
|-------------|--------------------------------------------------------------------------------------------------------|
| VLAN        | VLAN identifier of the VLAN.                                                                           |
| MAC Address | MAC address as learned from the source address field of a frame, or entered as part of a static entry. |
| Port/Vidx   | Port from which the MAC address was learned.                                                           |
| Status      | Whether the entry was a static entry or dynamically learned.                                           |
| daRoute     | Indicates whether received packets are switched on Layer 2 (0) or Layer 3 (1).                         |

**Example** To display the contents of the forwarding database, use the command:

```
sh swi fdb
```

**Related Commands** [enable switch learning](#)  
[show switch](#)  
[show switch filter](#)

## show switch filter

---

**Syntax** `SHoW SWItch FILter [Port={port-list|ALL}] [ActioN={FORward|DIScard}] [DEStaddress=macadd] [ENTRy=entry-list] [VLAN={vlan-name|1..4094}]`

where:

- *macadd* is an Ethernet six-octet MAC address, expressed as six pairs of hexadecimal digits delimited by hyphens.
- *entry-list* is an entry number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Entry numbers start at 0 and end at *m*, where *m* is the highest filter entry currently defined in the permanent forwarding database. Each port has its own permanent forwarding database.
- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.
- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.

**Description** This command displays information about some or all of the static switch filter entries ([Figure 7-29](#), [Table 7-19 on page 7-136](#)). The output can be limited to display only the entries matching the optional parameters.

The **action** parameter specifies whether frames matching the filter entry are forwarded or discarded.

The **entry** parameter must specify an existing filter entry or entries in the permanent forwarding database.

The **destaddress** parameter specifies the destination MAC address in the filter entry.

The **port** parameter specifies the outbound ports over which frames matching this filter entry are discarded or forwarded.

The **vlan** parameter specifies the numerical VLAN identifier with which the filter entry is associated.

Figure 7-29: Example output from the **show switch filter** command

| Switch Filters |               |                     |      |         |        |
|----------------|---------------|---------------------|------|---------|--------|
| Entry          | VLAN          | Destination Address | Port | Action  | Source |
| 0              | default (1)   | aa-ab-cd-00-00-01   | 1    | Forward | static |
| 1              | default (1)   | aa-ab-cd-00-00-02   | 1    | Forward | static |
| 0              | marketing (2) | aa-ab-cd-00-00-01   | 2    | Discard | static |
| 1              | marketing (2) | aa-ab-cd-00-00-02   | 2    | Discard | learn  |

Table 7-19: Parameters in output of the **show switch filter** command

| Parameter           | Meaning                                                                                                                                                                                                                                                                                                           |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Entry               | Number identifying the filter entry.                                                                                                                                                                                                                                                                              |
| Destination Address | Destination MAC address for the entry.                                                                                                                                                                                                                                                                            |
| VLAN                | VLAN name and identifier for the entry.                                                                                                                                                                                                                                                                           |
| Port                | Outbound port to match for the filter entry to be applied.                                                                                                                                                                                                                                                        |
| Action              | Whether the action specified by the filter entry is to forward or discard.                                                                                                                                                                                                                                        |
| Source              | This parameter is either "static" (indicating the filter is a static filter) or "learned" (indicating the filter is present either because it has been added with the <b>learn</b> parameter of the <b>set switch port</b> command, or has been dynamically learned during normal intrusion detection operation). |

**Examples** To display information about the entire permanent forwarding database, use the command:

```
sh swi fil po=all
```

To display information about the permanent forwarding database for port 3, use the command:

```
sh swi fil po=3
```

To display information about the permanent forwarding database for the *marketing* VLAN, use the command:

```
sh swi fil po=all vlan=marketing
```

To display which port MAC address 00-00-00-12-34-56 belongs to, use the command:

```
sh swi fil po=all dest=00-00-00-12-34-56
```

**Related Commands** [add switch filter action](#)  
[show switch](#)



## show switch hwfilter

**Syntax** `SHoW SWItch HWFilter[=filter-idlist]`

where *filter-idlist* is a single filter ID or a group as either a comma-separated list, range (specified as *n-m*), or a combination of the two. Filter IDs start at 1.

**Description** This command displays configuration information for the hardware-based packet filters (Figure 7-30, Table 7-20).

Figure 7-30: Example output from the **show switch hwfilter** command

```
Hardware-based Packet Filters
-----
Filter..... 6
  Rule Id..... 5
  Action..... COPY

  Rule Id..... 6
  Action..... COPY

Filter..... 8
  Rule Id..... 1
  Action..... DISCARD

  Rule Id..... 2
  Action..... DISCARD

  Rule Id..... 3
  Action..... DISCARD

  Rule Id..... 4
  Action..... DISCARD
```

Table 7-20: Parameters in output of the **show switch hwfilter** command

| Parameter | Meaning                                                                                                                                 |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Filter    | Filter ID for the hardware-based packet filter.                                                                                         |
| Rule Id   | Identifier for the classifier.                                                                                                          |
| Action    | Action associated with the classifier entry for the hardware-based packer filter. Options include discard, forward, setl2qos, and copy. |

**Examples** To display the configuration settings of switch filters, use the command:

```
sh swi hwf
```

**Related Commands** [add switch hwfilter](#)  
[delete switch hwfilter](#)

## show switch port

**Syntax** SHOW SWITCH PORT [= {*port-list* | ALL}]

where *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port.

**Description** This command displays general information about all ports or a specific one (Figure 7-31, Table 7-21 on page 7-139).

Figure 7-31: Example output from the **show switch port** command for port-based VLANs

```
Switch Port Information
-----
Port ..... 49
  Description ..... To intranet hub, port 49
  Status ..... ENABLED
  Link State ..... Up
  UpTime ..... 02:35:26
  Port Media Type ..... ETHERNET CSMACD
  Configured speed/duplex ..... Autonegotiate
  Actual speed/duplex ..... 1000 Mbps, full duplex
  MDI Configuration (Polarity) .. Not applicable
  Loopback ..... Off
  Configured master/slave mode .. Not applicable
  Actual master/slave mode ..... Not applicable
  Acceptable Frames Type ..... Admit All Frames
  Disabled egress queues ..... Q0, Q3-4
  BCast & MCast rate limit ..... 400 Kbytes/sec
  BCSC rate Limiting ..... Broadcast and Multicast enabled
  Egress rate limit ..... 10240 K/bs
  Learn limit ..... -
  Intrusion action ..... Discard
  Current learned, lock state ... 0, locked by thrashing
  Address learn thrash status ... Thrashing
  Address learn thrash action ... Disable Learning
  Address learn thrash timeout .. 1 second
  VLAN Status Trap ..... OFF
  Relearn ..... OFF
  Mirroring ..... Disabled
  Is this port mirror port ..... No
  Enabled flow control(s) ..... Pause
  VLAN(s) ..... default (1)
  Ingress Filtering ..... Off
  Trunk Group ..... -
  STP ..... default
  IGMP Filter ..... None
  Max-groups/Joined ..... Undefined/0
  IGMP Max-groups Action ..... Deny

  SFP vendor name ..... Allied Telesis
  SFP part number ..... AT-SPLX40
  SFP vendor SN ..... A02514N0410L000
  SFP date code ..... 05081300
  SFP type ..... 1000BASE-LX
  SFP length ..... 40km
  SFP wavelength ..... 1310nm
-----
```

Table 7-21: Parameters in output of the **show switch port** command

| Parameter                       | Meaning                                                                                                                                                |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| Port                            | Number of the switch port.                                                                                                                             |
| Description                     | Description of the port.                                                                                                                               |
| Status                          | Whether the port is enabled.                                                                                                                           |
| Link state                      | Whether the port is up or down.                                                                                                                        |
| Uptime                          | Hours:minutes:seconds of the elapsed time since the port was last reset or initialised.                                                                |
| Port Media Type                 | MAC entity type as defined in the ifType MIB object.                                                                                                   |
| Configured speed/<br>duplex     | Configured port speed and duplex mode for this port:<br>Autonegotiate                                                                                  |
|                                 | Speed 10Mbps<br>100Mbps<br>1000 Mbps<br>10 Gbps                                                                                                        |
|                                 | Duplex mode Half duplex<br>Full duplex                                                                                                                 |
| Actual speed/duplex             | Actual port speed and duplex mode for this port:                                                                                                       |
|                                 | Speed 10Mbps<br>100Mbps<br>1000 Mbps<br>10 Gbps                                                                                                        |
|                                 | Duplex mode Half duplex<br>Full duplex                                                                                                                 |
| MDI Configuration<br>(Polarity) | MDI/MDI-X polarity settings for this port.                                                                                                             |
|                                 | Configured MDI/<br>MDI-X polarity Auto (auto MDI/MDI-X)<br>Manual (auto MDI/MDI-X is disabled)<br>Not applicable                                       |
|                                 | Actual polarity (in<br>parentheses) for<br>the port MDI<br>MDI-X<br>Automatic<br>a dash when the polarity has not yet been<br>detected or is not known |
| Acceptable Frames Type          | Frame types that this port accepts:<br>Admit All Frames<br>Admit Only VLAN-tagged Frames                                                               |
| Disabled egress queues          | Egress queues that have been disabled. Traffic is discarded that<br>would normally be transmitted on these queues on these ports.                      |
| BCast & MCast rate limit        | Limit of the rate of reception of broadcast frames, or broadcast and<br>multicast frames, for this port, in Kbytes per second.                         |
| BCSC rate Limiting              | Whether rate limiting has been enabled for broadcast traffic,<br>enabled for broadcast and multicast traffic, or disabled.                             |
| Egress rate limit               | Maximum bandwidth to be transmitted out of this port, or trunk<br>group when the port is a member of a trunk group. Measured in<br>Kb/s.               |

Table 7-21: Parameters in output of the **show switch port** command (cont)

| Parameter                    | Meaning                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Learn limit                  | Number of MAC addresses that may be learned for this port. Once the limit is reached, the port locks out new MAC addresses. Either none or a number from 1 to 256.                                                                                                                                                                                       |
| Intrusion action             | Action taken on this port when a frame is received from an unknown MAC address when the port is locked:<br>Discard<br>Trap<br>Disable                                                                                                                                                                                                                    |
| Current learned, lock state  | Number of MAC addresses currently learned on this port and the state of locking. The current learned parameter is only incremented when there is a learn limit for the port. The lock state can be:<br>Not locked<br>Locked by limit<br>Locked by command<br>Locked by address thrashing                                                                 |
| Address learn thrash status  | The thrashing protection status of the port. If the thrash action is set to <b>vlandisable</b> , the status is shown for each VLAN that the port is a member of, with each VLAN listed on a separate line.                                                                                                                                               |
|                              | NotDetected      Thrashing has not been detected on the port.                                                                                                                                                                                                                                                                                            |
|                              | Thrashing          Thrashing has been detected and the specified thrash action has been applied.                                                                                                                                                                                                                                                         |
|                              | Disabled           Thrashing protection is disabled because the <b>thrashaction</b> is set to <b>none</b> .                                                                                                                                                                                                                                              |
| Address learn thrash action  | Trunked            The port is trunked and therefore thrashing protection is controlled by the trunk.                                                                                                                                                                                                                                                    |
|                              | Action taken when the address learn thrash limit is exceeded:                                                                                                                                                                                                                                                                                            |
|                              | Disable Learning      Address learning on the port is temporarily disabled.                                                                                                                                                                                                                                                                              |
|                              | Disable Port            The port is disabled, but the link remains up.                                                                                                                                                                                                                                                                                   |
|                              | Link Down              The port is disabled, and the link is down.                                                                                                                                                                                                                                                                                       |
| Address learn thrash timeout | Disable VLAN          The port is disabled for the VLAN on which thrashing is occurring.                                                                                                                                                                                                                                                                 |
|                              | The time, in seconds for which a port remains disabled after being disabled by thrashing protection. When a timeout value is specified and the port is currently disabled by the thrash limit, the time remaining before the port is re-enabled is shown in parentheses.<br>None                    The port remains disabled until manually re-enabled. |
| Enabled flow control(s)      | The flow control methods enabled for this port; one or more of "Pause" or "Jamming", or "-" if flow control is not enabled on the port.                                                                                                                                                                                                                  |
| VLAN Status Trap             | Whether an SNMP trap is sent when a port is enabled or disabled for the VLAN. Either <b>on</b> or <b>off</b> .                                                                                                                                                                                                                                           |
| Relearn                      | Whether dynamic MAC address learning is used on a port with a learn limit. When <b>off</b> , static is used.                                                                                                                                                                                                                                             |

Table 7-21: Parameters in output of the **show switch port** command (cont)

| Parameter                                                               | Meaning                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Mirroring                                                               | Whether traffic mirroring is enabled for this port. If mirroring is enabled, the following are also displayed:<br>The mirroring mode configured for this port; one of "None", "Rx" (for traffic received by this port), "Tx" (for traffic sent on this port), or "Both".<br>The port where mirrored frames are sent, or "no Mirror Port set" if a mirror port has not been set. |
| Is this port mirror port                                                | Whether this port is a mirror port.                                                                                                                                                                                                                                                                                                                                             |
| Enabled flow control(s)                                                 | Flow control method enabled for this port:<br>Pause<br>Jamming<br>not enabled                                                                                                                                                                                                                                                                                                   |
| VLAN(s)                                                                 | Name and VLAN identifier of the port-based VLANs to which the port belongs. Followed by "disabled" if the port has been disabled for that VLAN, for example by using the <b>disable switch port vlan</b> command, address learn thrash limiting, or QoS storm protection.                                                                                                       |
| Ingress Filtering                                                       | Whether ingress filtering is on.                                                                                                                                                                                                                                                                                                                                                |
| Trunk Group                                                             | Name of trunk group to which the port belongs, if any.                                                                                                                                                                                                                                                                                                                          |
| STP                                                                     | Name of the STP to which the port belongs.                                                                                                                                                                                                                                                                                                                                      |
| IGMP Filter                                                             | IGMP filter applied to the port, if any.                                                                                                                                                                                                                                                                                                                                        |
| Max-groups/Joined                                                       | The maximum number of multicast groups the port can join, if defined, and the number of multicast groups that the port is currently a member of.                                                                                                                                                                                                                                |
| IGMP Max-groups Action                                                  | The action to take—either deny or replace—when the port tries to join more multicast groups than the maximum allowed.                                                                                                                                                                                                                                                           |
| Cable Length                                                            | Approximate cable length used by this port in meters. Either <50m, 50-80m, 80-110m, 110-140m, >140m, or - (either port link is down, or port is operating at 10Mbps or 100Mbps). Cable length is reported for Gigabit Ethernet RJ-45 ports only. For AT-9900 switches only.                                                                                                     |
| <b>The following fields are displayed when a valid SFP is installed</b> |                                                                                                                                                                                                                                                                                                                                                                                 |
| SFP vendor name                                                         | Name of the SFP vendor.                                                                                                                                                                                                                                                                                                                                                         |
| SFP part number                                                         | Vendor part number or product name.                                                                                                                                                                                                                                                                                                                                             |
| SFP vendor SN                                                           | Vendor serial number.                                                                                                                                                                                                                                                                                                                                                           |
| SFP date code                                                           | Vendor manufacturing date code (2 digits each for year, month, day, and batch).                                                                                                                                                                                                                                                                                                 |
| SFP type                                                                | Port type for this SFP:<br>1000BASE-SX<br>1000BASE-ZX/CWDM<br>1000BASE-LX<br>1000BASE-T<br>100BASE-LX<br>100BASE-FX<br>BASE-BX10<br>OC-3<br>unknown                                                                                                                                                                                                                             |
| SFP length                                                              | Link length this SFP supports.                                                                                                                                                                                                                                                                                                                                                  |
| SFP wavelength                                                          | Wavelength at which this SFP transmits.                                                                                                                                                                                                                                                                                                                                         |

**Example** To display the configuration for switch port 1, use the command:

```
sh swi po=1
```

**Related Commands**

- [add vlan port](#)
- [delete vlan port](#)
- [disable switch port](#)
- [enable switch port](#)
- [reset switch port](#)
- [set switch port](#)

## show switch port counter

**Syntax** SHow SWItch POrt[={*port-list*|ALL}] COunter

where *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.

**Description** This command displays counters for a specific switch port or for all switch ports (Figure 7-32, Table 7-22 on page 7-143).

Figure 7-32: Example output from the **show switch port counter** command

```
Switch Port Counters
-----
Port 1. Ethernet MAC counters:
Combined receive/transmit packets by size (octets) counters:
 64                      5 512 - 1023                      0
 65 - 127                6 1024 - MaxPktSz                203924
128 - 255                0
256 - 511                0

General Counters:
Receive                      Transmit
Octets                      216975136 Octets                      852
Pkts                        203924 Pkts                        11
CRCErrors                  0
MulticastPkts              0 MulticastPkts              6
BroadcastPkts              0 BroadcastPkts              5
FlowCtrlFrms               0 FlowCtrlFrms               0
OversizePkts               0
Fragments                  0
Jabbers                     0
UpsupportOpcode            0
UndersizePkts              0
                               Collisions                      0
                               LateCollisions                 0
                               ExcessivCollsns                 0

Miscellaneous Counters:
MAC TxErr                  0
MAC RxErr                  0
Drop Events                 0
-----
```

Table 7-22: Parameters in output from **show switch port counter** command

| Parameter                                                   | Description                                                                                                             |
|-------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>Ethernet MAC counters</b>                                |                                                                                                                         |
| Combined receive/transmit packets by size (octets) counters | Number of packets in each size range received and transmitted.                                                          |
| 64                                                          | Number of 64 octet packets received and transmitted.                                                                    |
| 65 - 127                                                    | Number of 65 - 127 octet packets received and transmitted.                                                              |
| 128 - 255                                                   | Number of 128 - 255 octet packets received and transmitted.                                                             |
| 256 - 511                                                   | Number of 256 - 511 octet packets received and transmitted.                                                             |
| 512 - 1023                                                  | Number of 512 - 1023 octet packets received and transmitted.                                                            |
| 1024 - MaxPktSz                                             | Number of packets received and transmitted with size 1024 octets to the maximum packet length.                          |
| 1519 - 1522                                                 | Number of 1519 - 1522 octet frames received and transmitted.                                                            |
| <b>General Counters</b>                                     |                                                                                                                         |
| <b>Receive</b>                                              | Counters for traffic received.                                                                                          |
| Octets                                                      | Number of octets.                                                                                                       |
| Pkts                                                        | Number of packets.                                                                                                      |
| CRCErrors                                                   | Number of CRC error events.                                                                                             |
| MulticastPkts                                               | Number of multicast packets.                                                                                            |
| BroadcastPkts                                               | Number of broadcast packets.                                                                                            |
| FlowCtrlFrms                                                | Number of good Flow Control frames received.                                                                            |
| OversizePkts                                                | Number of oversize packets.                                                                                             |
| Fragments                                                   | Number of fragments.                                                                                                    |
| Jabbers                                                     | Number of jabber frames.                                                                                                |
| UnsupportOpcode                                             | Number of MAC Control frames with unsupported opcode (i.e. not Pause).                                                  |
| UndersizePkts                                               | Number of undersized packets.                                                                                           |
| <b>Transmit</b>                                             | Counters for traffic transmitted                                                                                        |
| Octets                                                      | Number of octets.                                                                                                       |
| Pkts                                                        | Number of packets.                                                                                                      |
| MulticastPkts                                               | Number of multicast packets.                                                                                            |
| BroadcastPkts                                               | Number of broadcast packets.                                                                                            |
| FlowCtrlFrms                                                | The number of Flow Control frames transmitted.                                                                          |
| Collisions                                                  | Total number of collisions seen by the MAC.                                                                             |
| LateCollisions                                              | Total number of late collisions seen by the MAC.                                                                        |
| ExcessivCollsns                                             | Number of frames dropped in the transmit MAC due to excessive collisions. This is applicable for Half-Duplex mode only. |
| <b>Miscellaneous Counters</b>                               |                                                                                                                         |

Table 7-22: Parameters in output from **show switch port counter** command (cont)

| Parameter  | Description                                                                                                                                                               |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Mac TxErr  | Number of frames not transmitted correctly or dropped due to internal MAC transmit error.                                                                                 |
| Mac RxErr  | Number of Receive Error events seen by the receive side of the MAC.                                                                                                       |
| DropEvents | Number of instances that the port was unable to receive packets due to insufficient bandwidth to one of the PP internal resources, such as the DRAM or buffer allocation. |

**Example** To display counters for switch port 1, use the command:

```
sh swi po=1 cou
```

**Related Commands** [set switch port](#)  
[show switch counter](#)  
[show switch port](#)

## show switch port intrusion

**Syntax** SHow SWItch POrt[={*port-list*|ALL}] INTRusion

where *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.

**Description** This command shows a list of MAC addresses for devices that are active on a port, but which are not valid devices allowed or learned for the port. The list contains entries if the **intrusionaction** parameter (with the **set switch port** command) is of the type **trap** ([Figure 7-33](#)).

The **port** parameter specifies the port for which to display the intrusion list. The default is **all**.

Figure 7-33: Example output from the **show switch port intrusion** command

```
Switch Port Information
-----
Port 2 -      13 intrusion(s) detected
  00-00-c0-1d-2c-f8  00-90-27-87-a5-22  00-00-cd-01-00-4a
  00-d0-b7-4d-93-c0  08-00-5a-a1-02-3f  00-d0-b7-d5-5f-a9
  00-b0-d0-20-d1-01  00-90-99-0a-00-49  00-10-83-05-72-83
  00-00-cd-00-45-9e  00-00-c0-ad-a3-d0  00-a0-24-8e-65-3c
  00-90-27-32-ad-61
-----
```

**Example** To display a list of MAC addresses for devices active on port 2, but which are not valid devices, use the command:

```
sh swi po=2 intr
```

**Related Commands** [set switch port](#)



## show switch trunk

**Syntax** `SHoW SWItch TRUnk [=trunk]`

where *trunk* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen.

**Description** This command displays information about the specified trunk group, or all trunk groups on the switch (Figure 7-34, Table 7-23).

The **trunk** parameter specifies the name of the trunk group. The name is not case sensitive. The name uniquely identifies the trunk group. The trunk group specified must already exist.

Figure 7-34: Example output from the **show switch trunk** command

```
Switch trunk groups
-----
Trunk group name ..... Uplink
Speed ..... 1000Mbps
Ports ..... 25,26
Address learn thrash status ..... Not Detected
Address learn thrash action ..... Disable Learning
Address learn thrash timeout ..... 1 second
Ports disabled by learn thrashing ... None
-----
```

Table 7-23: Parameters in output of the **show switch trunk** command

| Parameter                   | Meaning                                                                                                                                                                                                      |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Trunk group name            | Name of the trunk group.                                                                                                                                                                                     |
| Speed                       | Whether the configured speed of the trunk group ports is 10Mbps, 100Mbps, 1000Mbps, 10Gbps, or speed has not been set yet (-).                                                                               |
| Ports                       | Ports in the trunk group listed by port number.                                                                                                                                                              |
| Address learn thrash status | The thrashing protection status of the trunk. If the thrash action is set to <b>vlandisable</b> , the status is shown for each VLAN that the trunk is a member of, with each vlan listed on a separate line. |
|                             | NotDetected      Thrashing has not been detected on the trunk.                                                                                                                                               |
|                             | Thrashing      Thrashing has been detected and the specified thrash action has been applied.                                                                                                                 |
|                             | Disabled      Thrashing protection is disabled because the thrash action is set to none.                                                                                                                     |

Table 7-23: Parameters in output of the **show switch trunk** command

| Parameter                         | Meaning                                                                                                                                                                                                                                                                     |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Address learn thrash action       | Action taken when the address learn thrash limit is exceeded.                                                                                                                                                                                                               |
|                                   | Disable Learning      Learning on all ports in the trunk is temporarily disabled.                                                                                                                                                                                           |
|                                   | Disable Port      All ports in the trunk are disabled, but the links remain up.                                                                                                                                                                                             |
|                                   | Link Down      All ports in the trunk are disabled, and the links are down.                                                                                                                                                                                                 |
| Address learn thrash timeout      | Disable VLAN      All ports in the trunk are disabled for the VLAN on which thrashing is occurring.                                                                                                                                                                         |
|                                   |                                                                                                                                                                                                                                                                             |
| Address learn thrash timeout      | The time, in seconds for which a trunk remains disabled after being disabled by thrashing protection. When a timeout value is specified and the trunk is currently disabled by the thrash limit, the time remaining before the trunk is re-enabled is shown in parentheses. |
|                                   | If None is shown, the trunk remains disabled until manually re-enabled.                                                                                                                                                                                                     |
| Ports disabled by learn thrashing | A list of the ports in the trunk group that have been disabled by address learn thrashing, if the <b>thrashaction</b> is <b>portdisable</b> or <b>linkdown</b> .                                                                                                            |
|                                   | If Not applicable is shown, either the specified <b>thrashaction</b> is <b>learndisable</b> or <b>vlandisable</b> , or thrashing protection is disabled.                                                                                                                    |

**Example** To display information about all trunk groups, use the command:

```
sh swi tr
```

To display the settings for the Uplink trunk group, use the command:

```
sh swi tr=uplink
```

**Related Commands**

- [add switch trunk](#)
- [create switch trunk](#)
- [delete switch trunk](#)
- [destroy switch trunk](#)
- [set switch trunk](#)

## show vlan

**Syntax** `SHOW VLAN [= {vlan-name | 1..4094}]`

`SHOW VLAN [=ALL]`

where *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.

**Description** This command displays information about all VLANs or one, and varies depending on whether you have configured:

- port-based VLANs ([Figure 7-35](#))
- multiple VLANs ([Figure 7-36 on page 7-148](#))
- private VLANs ([Figure 7-37 on page 7-149](#))
- nested VLAN ([Figure 7-38 on page 7-149](#))

For a description of parameters, see [Table 7-24 on page 7-150](#).

Figure 7-35: Example output from the **show vlan** command (switch supports port-based VLANs only)

### VLAN Information

```

-----
Name ..... default
Identifier ..... 1
Status ..... static
Untagged ports ..... 1-24
Tagged ports ..... None
Disabled ports ..... None
Spanning Tree ..... default
Trunk ports ..... None
Mirror port ..... None
Broadcast limit .... Disabled
Multicast limit .... Disabled
Attachments:
Module           Protocol      Format      Discrim    MAC address
-----
GARP              Spanning tree 802.2      42         -
IP                IP            Ethernet    0800       -
IP                ARP           Ethernet    0806       -
-----

```

Figure 7-36: Example output from the **show vlan** command (switch supports multiple VLANs)

## VLAN Information

```

Name ..... fred
Identifier ..... 101
Status ..... static
Type ..... Multiple type
Private ..... No
Nested ..... No
Untagged ports ..... 1-8
Tagged ports ..... None

```

## Associations:

| IP Address    | Network Mask  | Ports |
|---------------|---------------|-------|
| 192.168.1.0   | 255.255.255.0 | 1-8   |
| 202.236.223.0 | 255.255.255.0 | 6-8   |

| Index | Encap. | Protocol Name | Ports |
|-------|--------|---------------|-------|
| 0     | EthII  | IPX           | 1-8   |
| 1     | EthII  | AppleTalk     | 1-4   |

```

Port associations .. None
Disabled ports..... None
Spanning Tree ..... default
Trunk ports ..... None
Mirror port ..... None
Broadcast limit .... Disabled
Multicast limit .... Disabled

```

## Attachments:

| Module | Protocol      | Format   | Discrim | MAC address |
|--------|---------------|----------|---------|-------------|
| GARP   | Spanning tree | 802.2    | 42      | -           |
| IP     | IP            | Ethernet | 0800    | -           |
| IP     | ARP           | Ethernet | 0806    | -           |

Figure 7-37: Example output from the **show vlan** command (switch supports private VLANs)

```

VLAN Information
-----
Name ..... vlan3
Identifier ..... 3
Status ..... static
Type..... Multiple type
Private ..... Yes
Nested ..... No
Untagged ports ..... 2-3,5,7-9,20-21
Tagged ports ..... None
Uplink ports ..... 20-21
Private ports ..... 2-3,5,7-9
Egress Filtering.... Enabled
Range Filtering .... OFF
Disabled ports..... None
Spanning Tree ..... default
Trunk ports ..... 20-21
Mirror port ..... None
Broadcast limit .... Disabled
Multicast limit .... Disabled
Attachments:
Module          Protocol      Format      Discrim      MAC address
-----
GARP            Spanning tree 802.2      42           -
IP              IP            Ethernet    0800         -
IP              ARP           Ethernet    0806         -
-----

```

Figure 7-38: Example output from the **show vlan** command (switch supports nested VLANs)

```

VLAN Information
-----
Name ..... vlan2
Identifier ..... 2
Status ..... static
Type..... Multiple Type
Private ..... No
Nested ..... Yes
Untagged ports ..... 1-2
Tagged ports ..... 3-4
Core ports ..... 3-4
Customer ports ..... 1-2
Associations ..... Port only
Port associations .. 1-2
Disabled ports..... None
Spanning Tree ..... default
Trunk ports ..... 3-4
Mirror port ..... None
Attachments:
Module          Protocol      Format      Discrim      MAC address
-----
GARP            Spanning tree 802.2      42           -
-----

```

Table 7-24: Parameters in output of the **show vlan** command

| Parameter         | Meaning                                                                                                                                                                                                          |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name              | Name of the VLAN.                                                                                                                                                                                                |
| Identifier        | Numerical VLAN identifier of the VLAN.                                                                                                                                                                           |
| Status            | Whether the VLAN is dynamic or static.                                                                                                                                                                           |
| Type              | VLAN classification rules supported.                                                                                                                                                                             |
| Private           | Whether the VLAN is private. In a private VLAN, all traffic received on a given port is sent to a predefined uplink port, regardless of the MAC Destination address or VLAN.                                     |
| Nested            | Whether nested VLAN functionality is enabled.                                                                                                                                                                    |
| Untagged Ports    | A list of untagged ports that belong to the VLAN.                                                                                                                                                                |
| Configured        | Specifies which ports are configured for the specified VLAN if the VLAN has ports that are either assigned to another VLAN, or configured for another VLAN but assigned to this VLAN by Dynamic VLAN Assignment. |
| Actual            | Specifies which ports are actually in the specified VLAN if the VLAN has ports that are either assigned to another VLAN, or configured for another VLAN but assigned to this VLAN by Dynamic VLAN Assignment.    |
| Tagged Ports      | A list of tagged ports that belong to the VLAN.                                                                                                                                                                  |
| Core ports        | For nested VLANs, ports that are connected to a service provider.                                                                                                                                                |
| Customer ports    | For nested VLANs, ports that are connected to customers requesting a service.                                                                                                                                    |
| Uplink ports      | For private VLANs, the network-facing port (or ports if a trunk), i.e. the port, or ports, through which a private VLAN accesses another network.                                                                |
| Private ports     | For private VLANs, a list of the customer-facing ports, i.e. switch ports that cannot communicate with each other, but can access another network.                                                               |
| Range Filtering   | VLAN Range Filtering supports a configurable maximum-allowed VLAN ID. Incoming VLAN tagged packets with a VLAN ID greater than this value are discarded and not learned.                                         |
| Associations      | The associations between the port and an IP subnet, protocol, limited protocol, or MAC address added to the VLAN. Or the association between the port and the Port VLAN classification rule of the VLAN.         |
| IP address        | IP address of the Ip subnet.                                                                                                                                                                                     |
| Network Mask      | Network for the Ip subnet.                                                                                                                                                                                       |
| Index             | Numerical index number representing the protocol, limited protocol, or MAC address entry.                                                                                                                        |
| Encap.            | Encapsulation of the frame; either EthII (IEEE 802.3), SAP (IEEE 802.2), SNAP (IEEE802.2 using the SNAP mechanism), or NetwareRaw (original Novell 802.2).                                                       |
| Protocol Name     | Descriptive name of the protocol or limited protocol. for limited protocols, the name is "IP", "IPX", or "Other".                                                                                                |
| Ports             | Ports that associations between the Ip subnet, protocol, limited protocol or MAC address have been made.                                                                                                         |
| Port associations | Ports that have been associated with the Port VLAN classification rule of the VLAN.                                                                                                                              |

Table 7-24: Parameters in output of the **show vlan** command (cont)

| Parameter          | Meaning                                                                                                                                                                                                                                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Disabled ports     | Ports that are members of the VLAN, but which have been disabled at switch level for example, using the <b>disable switch port</b> or <b>disable switch port vlan</b> commands.                                                                                                                                                              |
| Spanning Tree      | Name of the Spanning Tree Protocol to which the VLAN belongs.                                                                                                                                                                                                                                                                                |
| Trunk ports        | List of switch ports that belong to trunk groups.                                                                                                                                                                                                                                                                                            |
| Mirror port        | Mirror port for the switch, or "None".                                                                                                                                                                                                                                                                                                       |
| Broadcast limit    | Packet rate per second at which L2 broadcast packets are transmitted. Broadcast packets above this rate are dropped. If the limit is not <b>none</b> , the status of broadcast storm protection for the VLAN (enabled or disabled) is shown in parentheses. If the limit is <b>none</b> , broadcast storm protection is implicitly disabled. |
| Multicast limit    | Packet rate per second at which L2 multicast packets are transmitted. Multicast packets above this rate are dropped. If the limit is not <b>none</b> , the status of multicast storm protection for the VLAN (enabled or disabled) is shown in parentheses. If the limit is <b>none</b> , multicast storm protection is implicitly disabled. |
| <b>Attachments</b> | This section contains information about attachments to the VLAN made by other modules in the switch.                                                                                                                                                                                                                                         |
| Module             | Name of the software module attached to the VLAN.                                                                                                                                                                                                                                                                                            |
| Protocol           | Name of the protocol, which is determined from the format and discriminator.                                                                                                                                                                                                                                                                 |
| Format             | Encapsulation format specified by the module.                                                                                                                                                                                                                                                                                                |
| Discrim            | Discriminator specified by the module to identify which packets of the given format should be received.                                                                                                                                                                                                                                      |

**Examples** To display information on the Marketing VLAN, use the command:

```
sh vlan=marketing
```

To display information on all VLANs, use either of the commands:

```
sh vlan
```

```
sh vlan=all
```

**Related Commands** [create vlan](#)

## show vlan debug

**Syntax** SHow VLAN DEbug

**Description** This command displays debug information for all VLANs (Figure 7-39, Table 7-25).

Figure 7-39: Example output from the **show vlan debug** command

| Vlan  | Enabled Debug Modes | Output | Timeout |
|-------|---------------------|--------|---------|
| Vlan1 | PKT                 | 16     | NONE    |

Table 7-25: Parameters in output of the **show vlan debug** command

| Parameter           | Meaning                                                                                              |
|---------------------|------------------------------------------------------------------------------------------------------|
| VLAN                | String comprising the constant “Vlan” and the VLAN identifier of the VLAN.                           |
| Enabled Debug Modes | Whether the debugging option for the VLAN is PKT or none.                                            |
| Output              | Output device for the VLAN. Shown when a debug mode is presently enabled.                            |
| Timeout             | Time in seconds that debugging options for the VLAN are enabled. Shown when a debug mode is enabled. |

**Examples** To display debugging information for all VLANs, use the command:

```
sh vlan de
```

**Related Commands** [disable vlan debug](#)  
[enable vlan debug](#)



---

## show vlan port

---

**Syntax** `SHOW VLAN[={vlan-name|1..4094|ALL}] PORT[={port-list|ALL}]`

where:

- *port-number* is the number of a port on the switch.
- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.
- *port-list* is a single port number or a group as either a comma-separated list, range (specified as *n-m*), or a combination of the two. Port numbers start at 1 and end at *m*, where *m* is the highest numbered switch Ethernet port.

**Description** This command displays information about the VLAN to which the port belongs:

- public ([Figure 7-40 on page 7-154](#))
- nested ports ([Figure 7-41 on page 7-154](#))
- private port ([Figure 7-42 on page 7-155](#))

For a description of parameters, see [Table 7-26 on page 7-155](#).

The **vlan** parameter specifies the name of an existing VLAN or a numerical VLAN identifier (VID). The name is not case sensitive although case is preserved for display purposes. Specifying **all** or no value for this parameter means the same.

The **port** parameter specifies a port number or a group. If the **port** parameter contains a port number or a list of ports, and the **vlan** parameter is not specified, then all VLANs to which the port belongs are shown. If the **port** parameter contains a port number or a list of ports, and the **vlan** parameter is specified, then the port must belong to the specified VLAN. Specifying **all** or no value for this parameter means the same.

Figure 7-40: Example output from the **show vlan port** command

```

VLAN Port Information
-----
Port ..... 1
  VLAN Name ..... sales1 (235)
  Type ..... Multiple type
  Outgoing packets ..... untagged
  Associations:
  IP Address          Network Mask
  -----
  202.127.221.0      255.255.255.0
  202.128.167.0      255.255.255.0
  Index  Encapsulation  Protocol  Name
  -----
  0       EthII         0800      IP

  Port association ..... Yes

  VLAN Name ..... admin (105)
  Type ..... Multiple type
  Outgoing packets ..... untagged
  Associations:
  Index  Encapsulation  Protocol  Name
  -----
  0       EthII         6003      DEC DECNET
  1       EthII         6004      DEC LAT

  Port association ..... No

Port ..... 2
  VLAN Name ..... default (1)
  Type ..... IP subnet
  Outgoing packets ..... untagged
  Associations:
  Port association ..... Yes
-----

```

Figure 7-41: Example output from the **show vlan port** command for nested ports

```

VLAN Port Information
-----
Port ..... 3
  VLAN Name ..... default
  Type ..... Port-based
  Outgoing packets ..... untagged
  Associations ..... Port only
  Port association ..... Yes

  VLAN Name ..... vlan2
  Type ..... Multiple Type
  Nestedtype ..... CORE
  Outgoing packets ..... tagged
-----

```

Figure 7-42: Example output from the **show vlan port** command for a private port

```

VLAN Port Information
-----
Port ..... 3
  VLAN Name ..... vlan2 (private)
  Type ..... Multiple Type
  Port Type ..... Private
  Outgoing packets ..... untagged
  Associations ..... Port only
  Port association ..... Yes
-----

```

Table 7-26: Parameters in output of the **show vlan port** command

| Parameter        | Meaning                                                                                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Port             | Number of the switch port.                                                                                                                                                |
| VLAN Name        | Name of the VLAN followed by "(private)" if the VLAN is private. The default VLAN, VID=1, is named "default".                                                             |
| Type             | VLAN classification rules supported.                                                                                                                                      |
| Nestedtype       | Whether the type of nested port is Core or Customer.                                                                                                                      |
| Port Type        | For ports in a private VLAN, the type of the port; either "Private" or "Uplink".                                                                                          |
| Outgoing packets | Whether the packets transmitted out of this port for the VLAN are untagged or tagged.                                                                                     |
| Associations     | The associations between the port and an IP subnet and protocol added to the VLAN. Or the association between the port and the Port VLAN classification rule of the VLAN. |
| IP Address       | IP address of a subnet that is associated with the VLAN and port. Traffic from this subnet is classified as belonging to this VLAN.                                       |
| Network Mask     | IP mask of the subnet that is associated with the VLAN and port.                                                                                                          |
| Index            | Numerical index number representing the protocol IP subnet entry.                                                                                                         |
| Encapsulation    | Encapsulation of the frame; either "EthII" (IEEE 802.3), "SAP" (IEEE 802.2), "SNAP" (IEEE 802.2 using SNAP mechanism), or "NetwareRaw" (original Novell 802.3).           |
| Protocol         | Actual protocol value in hexadecimal.                                                                                                                                     |
| Name             | Name of the protocol or limited protocol.                                                                                                                                 |
| Port Association | Port VLAN classification rule of the VLAN to which the port is associated. Either yes or no.                                                                              |

**Examples** To display information about VLAN membership for all ports, use the command:

```
sh vlan po
```

**Related Commands**

- [add vlan port](#)
- [delete vlan port](#)
- [set vlan port](#)

