

Chapter 10

Interfaces

| | |
|--|-------|
| Introduction | 10-2 |
| Naming Interfaces | 10-2 |
| Simple Interface Names | 10-3 |
| Ethernet | 10-3 |
| Encapsulations | 10-4 |
| Configuration | 10-6 |
| Asynchronous Interfaces | 10-8 |
| Encapsulations | 10-8 |
| Configuration | 10-9 |
| Session Timeout | 10-12 |
| Connecting a Modem to the Asynchronous Port | 10-12 |
| MIB Counters | 10-13 |
| Autobauding | 10-14 |
| Making Asynchronous Ports Respond More Quickly | 10-14 |
| Testing Serial Data Circuits | 10-15 |
| Displaying Interfaces | 10-15 |
| Interface Link Traps | 10-16 |
| Managing Interfaces with SNMP | 10-16 |
| Command Reference | 10-17 |
| connect asyn | 10-17 |
| disable asyn | 10-17 |
| disable interface linktrap | 10-18 |
| enable asyn | 10-19 |
| enable interface linktrap | 10-19 |
| purge asyn | 10-20 |
| reset asyn | 10-20 |
| reset asyn counter | 10-21 |
| reset asyn history | 10-22 |
| reset eth | 10-22 |
| reset eth counters | 10-23 |
| reset interface counters | 10-23 |
| set asyn | 10-24 |
| set eth speed | 10-29 |
| set interface mtu | 10-30 |
| set interface traplimit | 10-31 |
| show asyn | 10-32 |
| show eth configuration | 10-40 |
| show eth counters | 10-41 |
| show eth macaddress | 10-47 |
| show eth receive | 10-48 |
| show eth state | 10-49 |
| show interface | 10-51 |

Introduction

This chapter describes how to configure, control and monitor interfaces, and the encapsulations supported on each interface. The chapter also describes the format of the Ethernet frame, and the naming conventions that are available for different interface types.

The interface described are:

- Ethernet
- asynchronous

The term *interface* refers to one of the physical ports on the switch. The physical ports connect the switch to a network, and all data enters and leaves the switch via the interface.

Ethernet interfaces described in this chapter are Ethernet ports (labelled ETH), and not the switch ports described in [Chapter 7, Switching](#). The main distinction between models is the combination of different types and numbers of interfaces. See the Hardware Reference for details of the interfaces available on each model.

Asynchronous ports can be used to connect terminals, printers and terminal ports on host computers. See [Chapter 46, Terminal Server](#) for information about using asynchronous ports for terminal serving. See [Chapter 47, Stream Printing for AT-9900 Switches](#) for information about using asynchronous ports for print serving using the stream printing service. Each frame of data includes a header that informs a receiving switch about the protocol carried in the frame. This header is specified by a set of rules referred to as an *encapsulation*. Some interface types can be used with more than one encapsulation. It is important to know about encapsulations for two reasons. Firstly, the information can be useful in debugging network problems, if traces of the packets being transmitted or received on a particular interface can be obtained. Secondly, information about encapsulations can be used to determine whether the switch can interoperate with other vendors' switches, since this depends on both switches supporting the same encapsulation(s) for a particular protocol.

Naming Interfaces

Commands that configure an interface or attach a routing module to use a particular interface, must specify the interface by name. Typically, commands use the **interface=interface** or **over=interface** parameter to specify an interface.

The asynchronous port is called *asyn0*; the Ethernet out-of-band management port is called *eth0*.

Simple Interface Names

Create simple interface names on x900-24X switches by concatenating the interface type with the interface instance. The interface type is an abbreviation of the full name of the interface. The instance is a non-negative number. The following table describes names for types of interfaces.

| Interface Type | Description |
|-----------------|---|
| Logical | |
| VLAN | Virtual LAN interface |
| PPP | Point-to-Point Protocol interface |
| Physical | |
| ASYN | Asynchronous interface |
| ETH | Ethernet management interface |
| PORT | Ethernet switch ports (including uplinks) |

For logical interfaces, the instance number is the module instance number specified in the **add** or **create** command for that module. Instance numbers may be chosen arbitrarily but common practice is to assign them sequentially, starting with 0.

For physical interfaces, the instance number is the physical port number, which the system determines. Physical ports are numbered from left to right as viewed, starting at 0.

The following table shows examples of valid names for simple interfaces on x900-24X switches.

| Interface name | Description |
|----------------|------------------------------------|
| eth0 | Ethernet port 0 |
| asyn4 | Asynchronous port 4 |
| port3 | Switch port 3 |
| ppp1 | Point-to-Point Protocol instance 1 |
| vlan1 | Virtual LAN 1 |

Ethernet

Ethernet encapsulation is used on *switch ports* on the switch. For more information about switch ports and VLAN tagging in Ethernet frames, see [Chapter 7, Switching](#).

Ethernet is a term that describes a particular family of interface types and encapsulations. Other common terms are *802.3* and *CSMA/CD*. Various physical media can carry Ethernet, including thin and thick coaxial cable, twisted pair wires, and optical fibre.

All these forms of Ethernet are characterised by these common features:

- A single medium carries all incoming and outgoing traffic.
- A number of stations may use the same medium for communicating with all other stations on the medium. All stations can see all the traffic on the medium.

- Stations wait for the medium to become free before attempting to send data on it. If more than one station attempts to send data simultaneously a collision results and the data being sent becomes invalid.
- Stations can be connected to or disconnected from the medium without disturbing the other stations on the medium.
- The order in which stations are attached to the physical medium is not important.

Ethernet runs at speeds of 10 Mbps, 100 Mbps, 1 Gbps, or 10 Gbps.

The out-of-band 10/100/1000 Mbps Ethernet port on x900-24X switches is dedicated to management traffic. This port is reserved for management **only**; the switch does not transmit frames between this port and switch ports.

Ethernet is used primarily to provide local area networking rather than wide area networking. The installation of Ethernet media within premises is normally the responsibility of the user of the premises rather than the telecommunications provider.

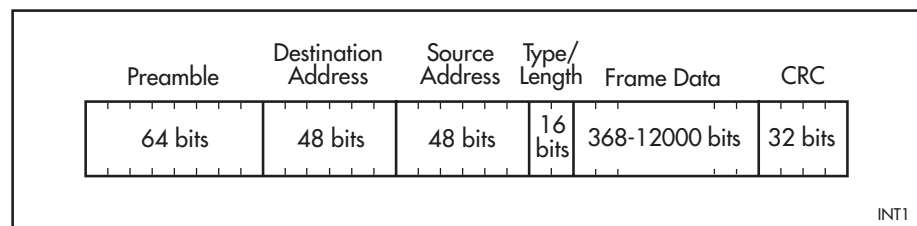
Ethernet was first defined in 1982. The original definition is generally referred to as Type 1 Ethernet and although it differs slightly from the modern standard, it is not very common today. Subsequent standards defined Type 2 Ethernet, which was largely ratified unchanged by the IEEE as IEEE 802.3.

Ethernet interfaces on the switch are specified by the IEEE Standard 802.3 or ISO 8802-3 standards. This is the standard used by most implementations. The switch physically supports all three versions of Ethernet, and is supplied with Type 2/ 802.3 selected.

Encapsulations

Since Ethernet is a single wire used by many stations at once and with many different protocols, encapsulation of protocol types is used to distinguish the protocols. Ethernet has been developed over a period of time, and the efforts of the Standards bodies following on from the vendors that developed Ethernet have led to different encapsulation types for Ethernet.

The following figure shows an Ethernet frame, which consists of a preamble followed by the data, and terminated with a CRC



The data begins with the station addresses of the receiver and sender of the frame. These address fields are both 6 octets long. Following the addresses is a 2-octet field, referred to here as the type/length field, that contains either a type field or a length.

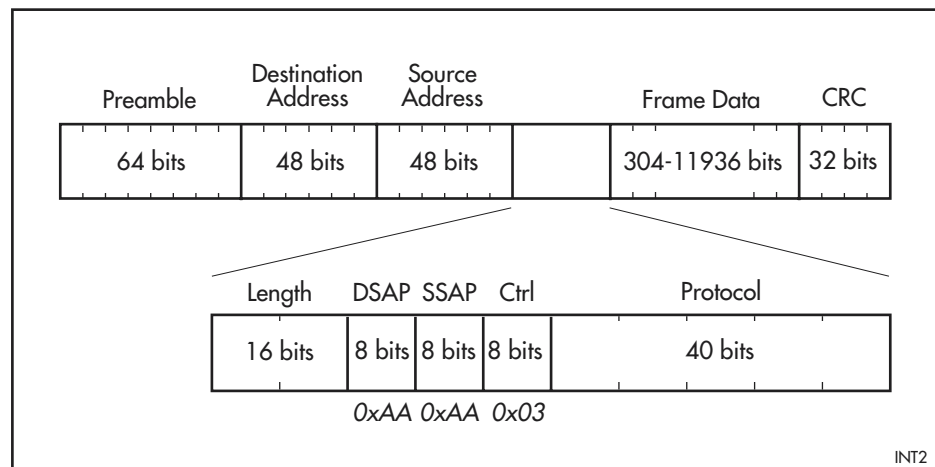
The type/length field was introduced by the vendors that developed Ethernet and was used to contain a protocol type. Different values in the type field distinguished different protocols. The values that are contained in this field are administered by Xerox Corporation and vendors of network equipment may apply to reserve a type field to define vendor-specific protocols.

The original vendor specifications were extended by the IEEE. This body developed standards in local area networking, including Ethernet. The Ethernet addresses and type/length field appear in the IEEE standards as part of the Ethernet-specific standard, IEEE Standard 802.3. Another standard, IEEE Standard 802.2, specifies the format of the frame after the type/length field. Since IEEE Standard 802.2 applies to other LAN media, such as Token Ring and FDDI, the frame after the type/length field cannot contain anything specific to Ethernet. For this reason the type/length field is used to specify a length, and is, in fact, the length of the rest of the frame.

Although there may appear to be a conflict between the use of the type/length field for both a frame type and a length, in practice there is no conflict. The maximum length of an Ethernet frame (including the preamble, addresses and the type/length field) is 1514 octets, so the maximum value of the type/length field as a length is 1500 octets. Ethernet types are assigned values greater than 1500. In the early days of Ethernet, some protocol types were assigned values below 1500, but these have since become obsolete.

When the IEEE introduced the standard that replaced the type field with a length field, parts of the networking community still wanted a way to specify that a particular Ethernet frame was a certain protocol type, without having to implement all of the IEEE Standard 802.2.

IEEE Standard 802.2 defines the two octets after the type/length field as *Service Access Points*, or SAPs, one for the source of the packet and one for the destination. A special SAP value (0xAA or 170 decimal) was defined to indicate that the packet containing this SAP value would use the *SubNetwork Access Protocol* (SNAP) mechanism. In IEEE Standard 802.2, the one or two octets after the SAPs are defined as the control field. For the SNAP format, this is defined as the single octet 0x03, used to indicate an “unnumbered information” frame. The SNAP format then defines the next 5 octets as a protocol type. Values in this field define the different protocols. The following figure shows the format of an Ethernet frame with SNAP encapsulation.



The switch supports the following encapsulation formats:

- Ethernet – type/length field used as a type
- 802.2 – use of IEEE Standard 802.2 with SAPs
- SNAP – use of the SNAP SAP
- Novell (referred to by Novell as *raw 802.3*)—802.2 format packet with destination and source SAPs of 0xFF, but without the other fields of a true 802.2 header (x900-48FE and AT-9900 only)

For the correct operation of a software module, Ethernet drivers must receive packets with the appropriate encapsulation and forward them to that module. The packets are specified by an encapsulation format and a discriminator. The following table lists discriminators for each encapsulation format.

| Format | Discriminator | Length (octets) |
|----------|--------------------|-----------------|
| Ethernet | Ethernet type | 2 |
| 802.2 | Destination SAP | 1 |
| SNAP | SNAP discriminator | 5 |
| Novell | - | - |

When a module specifies that the Ethernet drivers receive packets with the Novell format, a discriminator is not required for x900-48FE switches.

Configuration

An Ethernet interface on the x900-24X is automatically configured by the software modules when it starts. No user configuration of the Ethernet interfaces is required, except to enable software modules to use the interface. This is achieved by adding a software module interface and using the following clause:

```
interface=ethn
```

n is the number of the Ethernet interface being configured.

The modules in the x900-24X that are configured to use an Ethernet interface, and the encapsulations used on an interface, can be displayed with the following command:

```
show eth=n configuration
```

n is the number of the Ethernet interface.

A feature of Ethernet is the ability to send packets to more than one station at a time, using multicast addresses. The multicast addresses required by a software module are automatically entered into the list of receive addresses by that module. No action by the user is required.

Note that the list includes the broadcast address and any unicast addresses specified by the software modules that have configured to the Ethernet interface. Unicast addresses are distinguishable from multicast addresses by their first octet. The first octet of a unicast address is even, whereas for a multicast address it is odd. The broadcast address is a special multicast address that is received by all stations on an Ethernet. The switch is always configured to receive broadcast packets, even if no software modules are using the interface, so the list always includes the broadcast address.

The default MAC address used by the Ethernet interface can be displayed with the command:

```
show eth[=n] macaddress
```

where *n* is the number of the Ethernet interface being configured.

The addresses that the switch is configured to receive can be displayed with the command:

```
show eth[=n] receive
```

where *n* is the number of the Ethernet interface being configured.

The switch maintains a number of counters for each Ethernet interface. These counters are objects in three standard MIBs and the switch's enterprise MIB. For more information on MIBs, see [Chapter 39, Simple Network Management Protocol \(SNMP\)](#).

The counters are grouped into categories depending on the MIB to which they belong. The following table lists categories maintained for Ethernet interfaces.

| Category | Group | MIB table | RFC |
|------------|--------------------|----------------------|------|
| INTERFACE | Interfaces | Interfaces | 1213 |
| INTERFACE | Generic interfaces | Interface extensions | 1229 |
| DOT3STAT | Transmission | Dot 3 statistics | 1398 |
| COLLISION | Transmission | Collision statistics | 1398 |
| DIAGNOSTIC | Enterprise MIB | General | - |
| DIAGNOSTIC | Enterprise MIB | SONIC | - |

Counters from each of these four categories are displayed with the command:

```
show eth[=n] counter[=category]
```

where *n* is the number of the Ethernet interface and *category* is one of the four categories of counter. If a category is not specified, all categories are displayed.

The counters in each category may be cleared to zero with the command:

```
reset eth[=n] counter[=category]
```

where *n* is the number of the Ethernet interface and *category* is one of the four categories of counter. If a category is not specified, all counters are cleared.

Using the [reset eth counters command on page 10-23](#) to clear the counters does not clear the MIB counters themselves. Instead, the MIB counter contents are copied to offset storage locations that are subtracted from the MIB counters before being displayed by the [show eth counters command on page 10-41](#).

The state of the connection between the Ethernet interface and network can be displayed with the following command:

```
show eth[=n] state
```

n is the number of the Ethernet interface. The command shows whether there is a working link between the Ethernet interface and the Ethernet and, if so, the link's capabilities.

The Ethernet interfaces on the switch can be reset with the following command:

```
reset eth=n
```

n is the number of the Ethernet interface being reset. The Ethernet interface must be specified. A complete reset of the Ethernet interface is carried out with this command.

Important Data being sent or received when the Ethernet interface is reset is lost.

100Mbps Ethernet interfaces normally negotiate automatically with the network to decide which link speed (bit rate and duplex mode) to use. This allows 100Mbps Ethernet interfaces to operate with both 10Mbps and 100Mbps networks. If desired, auto-negotiation can be disabled and a particular link speed can be specified for a 100Mbit Ethernet interface with the following command:

```
set eth=n speed={autonegotiate|10mhalf|10mfull|100mhalf|100mfull}
```

Important Setting an inappropriate link speed can interfere with the operation of other devices on the network.

Asynchronous Interfaces

All models of the switch have at least one asynchronous interface, or port. This is a standard RJ45, DB9 male or DB9 female connector wired as a DTE (*Data Terminating Equipment*) interface. The asynchronous ports are identified by number, and are numbered sequentially starting from 0. The first interface is called asyn0.

All asynchronous ports use the RS-232C standard. At least four modem control lines are provided with each interface, and these are normally used as DTR, RTS, CTS, and CD. More information about asynchronous ports is in the Hardware Reference for the switch.

Asynchronous ports are normally used to connect terminals or modems to the switch. The cable types required to do this are described in the Hardware Reference. In general, most VT100-compatible terminals require a *crossed* cable (DTE-to-DTE).

Important The term *crossed* refers to the fact that the data pins (TxD and RxD) on the connector at one end of the cable are connected to the opposite pins (RxD and TxD respectively) on the connector at the other end of the cable. This is necessary because both the terminal and the switch have DTE interfaces.

An asynchronous port can be configured as a stream printer port using the stream printing service. See [Chapter 47, Stream Printing for AT-9900 Switches](#) for more information. To use stream printing, IP must be enabled and configured on all switches in the network providing access to the stream printer. See [Chapter 13, Internet Protocol \(IP\)](#) for more information.

Asynchronous ports may also be used as network interfaces.

Encapsulations

By default, no encapsulation is used on asynchronous ports. Data is transmitted and received as a clear character stream. This is appropriate for remote terminal or terminal emulation access and remote printing facilities.

The switch also supports the SLIP (Serial Line Internet Protocol) encapsulation, and CSLIP (Compressed SLIP using Van Jacobson's header compression) encapsulation.

SLIP encapsulation is used when the port is assigned to the IP module as an IP interface with the command:

```
add ip interface=slipn ...
```

where *n* is the number of the asynchronous port. See [Chapter 13, Internet Protocol \(IP\)](#) for more details about creating IP interfaces.

Configuration

Each asynchronous port can be individually configured to suit a wide range of different terminal types. The characteristics of a port can be changed by using the command:

```
set asyn=asyn-number option
```

Options for each port are listed in the following table.

Asynchronous port options

| Option | Description |
|----------------|--|
| Attention | Sets the attention character used to return from a virtual terminal session to the switch prompt. |
| Cdcontrol | Controls the way the switch interprets the state of the DCD input signal. If cdcontrol is set to ignore the switch ignores the state of the DCD input signal. If cdcontrol is set to connect the switch terminates existing connections when the DCD signal is deasserted (i.e. when a modem disconnects). If cdcontrol is set to online and the interface is configured as a printer port, output is not sent to the interface unless the DCD signal is asserted. |
| Databits | Sets the number of data bits per character transmitted by the port. |
| Defaultservice | Configures the port to automatically connect to a service when a user types anything at the terminal or an attached modem asserts DCD. |
| Dtrcontrol | Controls the way the switch controls the state of the DTR output signal. If dtrcontrol is set to connect the switch asserts DTR for the duration of the connection. If dtrcontrol is set to on or off , the DTR line is driven to the designated state. |
| Echo | Enables or disables the echoing of each character entered at a terminal. |
| Flow | Sets the flow control mechanism used for the port in both the receive and transmit directions. If flow is set to none the switch ignores all incoming flow control characters and lead transitions. If flow is set to character the switch uses xon/xoff flow control. If flow is set to hardware the switch uses the rts/cts lines for flow control. |
| Inflow | Sets the flow control mechanism used for the port in the receive direction. If flow is set to none , the switch ignores all incoming flow control characters and lead transitions. If flow is set to character , the switch uses xon/xoff flow control. If flow is set to hardware , the switch uses the rts/cts lines for flow control. |
| History | Sets the number of commands saved for command line recall. |
| Idletimeout | Specifies a period of time, in seconds, for a terminal connection's dedicated TTY device idle timer. If the specified time period lapses since the last time the dedicated TTY device received data from the client, the connection is terminated, and the terminal screen displays the login prompt. If 0 or off are specified, the idle timer remains off, and the session must be explicitly terminated. |
| Ipaddress | Sets the IP address associated with the port. This parameter may be required when the port is used as a network interface using SLIP or PPP. |
| Ipxnetwork | Sets the IPX network number associated with the port. This parameter may be required if the port is used as a network interface using PPP. |
| Login | Enables or disables the ability to log into the asynchronous port. |
| Maxoqlen | Sets the maximum number of character buffers that are permitted on the transmit queue for the port. |

Asynchronous port options

| Option | Description |
|---------------|--|
| Mtu | Sets the Maximum Transmission Unit (MTU), which is the maximum number of bytes per packet that may be transmitted by the port when it is used as a network interface. |
| Name | Assigns a text string used to identify the port, such as the name of the person whose terminal is normally connected to the port, or where the terminal is located. |
| Outflow | Sets the flow control mechanism used for the port in the transmit direction. If flow is set to none , the switch ignores all incoming flow control characters and lead transitions. If flow is set to character , the switch uses XON/XOFF flow control. If flow is set to hardware , the switch uses the RTS/CTS lines for flow control. |
| Page | Sets the number of lines of output displayed on the terminal before the switch pauses and waits for the user to press a key to continue. |
| Parity | Sets the parity of each character transmitted by the port. |
| Prompt | Sets the prompt to a string, the default prompt, or disables the prompt. |
| Secure | Controls whether a user must log in to the port before switch commands can be accepted. See Chapter 29, User Authentication for information about defining users and logging in to the switch. |
| Service | Allocates the port to be a host port for a named service. |
| Speed | Sets the speed of the port, from 75 bps to 115200 bps. The terminal and port must be set to the same speed. Autobauding is also available, provided the attention character used is set to [Break]. In this mode the port automatically adjusts to the speed of the terminal that is attached, up to 19200 bps. |
| Stopbits | Sets the number of stop bits per character transmitted by the port. |
| Tentimervalue | The period, in milliseconds, over which the port bundles characters, when the port is in ten mode. |
| Type | Sets the terminal type to vt100 or dumb . A dumb terminal is used for printing or terminals that do not support VT100 escape sequences. |

Asynchronous port defaults

Asynchronous ports are initially configured with default values listed in the following table.

| Option | Default setting |
|----------------|-----------------|
| Attention | break |
| Cdcontrol | ignore |
| Databits | 8 |
| Defaultservice | false |
| Dtrcontrol | on |
| Echo | on |
| Flow | hardware |
| History | 30 |
| Idletimeout | 0 |
| Inflow | hardware |
| Ipaddress | none |
| lpxnetwork | none |

| Option | Default setting |
|---------------|------------------|
| Login | on |
| Maxoqlen | 0 (Unrestricted) |
| Mtu | 1500 |
| Name | asyn # |
| Outflow | hardware |
| Page | 22 |
| Parity | none |
| Prompt | default (CMD>) |
| Secure | on |
| Service | none |
| Speed | 9600 |
| Stopbits | 1 |
| Tentimervalue | 100 |
| Type | vt100 |

To display the complete configuration for a particular asynchronous port, use the command:

```
show asyn=asyn-number
```

To display the complete configuration for all asynchronous ports, use the command:

```
show asyn=all
```

To display summary details for a particular asynchronous port, use the command:

```
show asyn=asyn-number summary
```

To display summary details for all asynchronous ports, use the command:

```
show asyn=all summary
```

The switch maintains a separate command history list for each asynchronous port, containing the last commands entered at the port. To display the history list, use the command:

```
show asyn=asyn-number history
```

Session Timeout

If you disable an asynchronous port, users can still log into the port but will be logged out if the session is idle for a configurable length of time. Users can log in to the disabled port by sending it a break signal. To configure this timeout functionality:

1. Set the timeout period.

If you require a different timeout than the default of 60 seconds, use the command:

```
set asyn [enable=break] timeout=1..65535
```

The timeout only applies if **enable=break**, which is its default.

2. Manually disable the port.

Use the command:

```
disable asyn
```

Note that if you are logged into an asynchronous port to manage the switch, you cannot disable that port by typing the disable command. You have to run the disable command from a script or from a different session, such as a telnet session.

To log into the asynchronous port:

- Connect as normal to the port through a terminal emulator or modem. Then send a break signal to get a log in prompt. The method of sending a break signal depends on the terminal application.
- To start a new session after the port has timed out, send a break signal. This enables the port, which will then provide you with a log in prompt.

Connecting a Modem to the Asynchronous Port

If a modem is connected, configure the switch to make and/or accept calls via the modem. To set the **cdcontrol** parameter to **connect** and the **flow** parameter to **hardware**, enter the command:

```
set asyn cdcontrol=connect flow=hardware
```

If the terminal or modem is used with communications settings other than the default settings, then configure the asynchronous port to match the terminal or modem settings by using the **set asyn** command.

A port connected to a modem should always be set to a fixed speed matching that of the modem.

MIB Counters

The switch maintains a number of counters for each asynchronous port. The counters are objects in two standard MIBs and the switch's enterprise MIB. Counters are grouped into categories depending on the MIB to which they belong. The following table lists the categories maintained for asynchronous ports.

| Category | Group | MIB table | RFC |
|------------|----------------|------------------------|------|
| INTERFACE | Interfaces | Interfaces | 1213 |
| RS232 | Transmission | Asynchronous port | 1659 |
| DIAGNOSTIC | Enterprise MIB | Asynchronous interface | - |

For more information about SNMP and MIBs, see [Chapter 39, Simple Network Management Protocol \(SNMP\)](#).

To display the MIB counters for an asynchronous port, use the command:

```
show asyn [=n] counter [=category]
```

where *n* is the number of the asynchronous port and *category* is one of the three counter categories. If a category is not specified, all categories are displayed.

Objects from the general input and output signal tables (see RFC 1659) are displayed by the **show asyn** command.

To clear counters in each category to zero, use the command:

```
reset asyn [=n] counter [=category]
```

where *n* is the number of the synchronous port and *category* is one of the three counter categories. If a category is not specified, all counters are cleared.

Using the **reset asyn** command to clear the counters does not clear the MIB counters themselves. Instead, the MIB counter contents are copied to offset storage locations that are subtracted from the MIB counters before being displayed by the **show asyn** command.

To enable or disable each asynchronous port, use the commands:

```
enable asyn=n
disable asyn=n
```

where *n* is the number of the asynchronous port. When an asynchronous port is disabled it does not transmit or receive data. When the port is enabled, all configuration parameters are restored to the settings in effect prior to the port being disabled. The default state of an asynchronous port is enabled.

Important Data being received or transmitted when the asynchronous port is disabled or reset is lost.

To reset an asynchronous port, use the command:

```
reset asyn=n
```

where *n* is the number of the asynchronous port. Any current connections are disconnected and the configuration parameters are restored from nonvolatile storage.

To reset the command history, use the command:

```
reset asyn history
```

The specific commands to change the parameters of a particular asynchronous port are given in “[Command Reference](#)” on page 10-17. As an example, to change the name of port 6 to “test” and the speed to 9600 bps, use the command:

```
set asyn=6 name=test speed=9600
```

All port configuration parameters are held in non-volatile memory and are retained over a power cycle.

Autobauding

Asynchronous ports may be set to autobauding mode. In this mode the switch adjusts the speed of the port to match the speed of the terminal attached to the port, up to a maximum speed of 19200 bps. For autobauding to work, the user should always press the [Enter] or [Return] key on the terminal several times until the switch prompt appears on the screen. At this point the switch has set the speed of the port. If a key other than [Enter] or [Return] is pressed while the switch is setting the port speed, the speed may be incorrectly set. In this case, there is no response from the switch or “garbage” characters appear on the terminal screen. To fix this, press [Break] two or more times, followed by [Enter] or [Return] several times.

Some terminals require the [Break] key to be held down for about a second to properly send a [Break]. Additionally, some terminals require a brief pause between multiple [Break]s.

Once the speed is set on an autobauding port, the switch does not change it unless one of the following events occurs:

- The switch is turned off.
- [Break] is pressed twice, in which case the switch “forgets” the current speed and waits for [Enter] or [Return] to be pressed several times to set the speed again.
- The terminal is switched off. This sometimes has the effect of sending [Break]s to the switch.

Making Asynchronous Ports Respond More Quickly

When an asynchronous port is in *ten mode*, it bundles together the characters that it receives within a certain time period, instead of passing them one at a time to a higher protocol layer for processing. The time period over which characters are bundled is set by the *ten timer*.

Bundling reduces the load on the CPU by spreading the character processing overhead across several characters. If a remote terminal session is involved, bundling also reduces the number of packets on the network by sending more characters in each packet. However, bundling reduces terminal responsiveness.

A ten timer value of 100 milliseconds is generally a good compromise between responsiveness and processing overhead. If you need to increase the port’s responsiveness, you can reduce the length of the ten timer, by using the command:

```
set asyn[=port-number] tentimervalue=20..100 [other optional  
parameters]
```

Unless you are logged in via the port you want to change, also specify the asynchronous port number.

The default **tentimervalue** is 100 milliseconds.

Testing Serial Data Circuits

Wide area data circuits are normally leased from the Telecom supplier. A point-to-point circuit has an NTU or modem at each end. These normally allow some limited testing of the circuit to be done. Unfortunately, there are a large number of different types of NTU and modem, so it is not possible to predict the exact functionality. The following gives an indication of the basic features common to most modems and NTUs.

In the remainder of this section, the term 'NTU' is used exclusively.

- Carrier detect** This signal is normally available at the data interface of the NTU as well as being shown on a front panel LED. It must be present for the NTU to operate correctly. If this fails, it usually means that the data circuit is faulty or the NTU at the other end is not functioning. In either case, the Telecom supplier should be called to fix the problem. Some other possible names for this signal include RLSD, 109, CD and EQG. If the circuit quality is poor, this signal may have frequent short transitions. This results in poor link throughput.
- Loopback** This feature is not normally present as an indicator, but rather as one or more front panel buttons, sometimes associated with an LED to show that the NTU is in a test mode. The loopback functionality available varies from NTU to NTU, depending on the type, and exactly what has been selected at installation time. Loopbacks allow the data circuit to be tested in stages, by progressively looping back first the local NTU and then the remote NTU. If for instance, the remote NTU loopback fails, but the local loopback is successful, it indicates a fault in either the data circuit or the remote NTU and the Telecom supplier should be notified. If the remote and local tests are successful, it indicates that the problem is either in the remote NTU or the network equipment at the remote end. The tests should be reversed from the remote end to eliminate the remote NTU.
- Data indicators** These are front panel mounted LEDs on the NTU and can be used to see that data is flowing in both directions.

Displaying Interfaces

The switch stores information about interfaces as objects in the Interfaces Table of MIB-II, defined in RFC 1213 *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*. To display the contents of the Interfaces Table, use the command:

```
show interface
```

To display detailed information about a specific interface, use the command:

```
show interface={ifindex|interface}
```

where *ifIndex* is the index of the interface in the Interfaces Table and *interface* is the interface name.

To display counters for all the interfaces, use the command:

```
show interface counter
```

For a detailed description of the objects in the Interfaces Table of MIB-II, see [Appendix C, SNMP MIBs](#).

Interface Link Traps

When an interface changes to or from the “Down” state, an SNMP trap can be sent to any SNMP manager stations (trap hosts) that have been defined.

The general operation of link traps is defined in RFC 1157, *Simple Network Management Protocol*. In the typical multi-layered interface environment, each protocol layer for which an interface entry exists in the interface table can generate link up/down traps.

Since interface state changes tend to propagate through the protocol layers, multiple traps may be generated as the result of a single link failure. RFC 1573, *Evolution of the Interfaces Group of MIB-II*, resolves this issue by providing a mechanism for enabling and disabling link trap generation on a specific interface. This allows stacked interfaces to be configured so that only one trap is sent for a link transition.

Link traps are disabled by default on the switch. Link traps can be enabled or disabled on a per-interface basis by using the commands:

```
enable interface linktrap
disable interface linktrap
```

To display current settings for link traps, use the command:

```
show interface
```

The potential exists in a large or busy network for a high volume of trap messages to be generated. To set the maximum number of link traps generated per minute for each static interface or for all dynamic interfaces, use the command:

```
set interface traplimit
```

Managing Interfaces with SNMP

Switch interfaces can be enabled or disabled via SNMP by setting the *ifAdminStatus* object in the *ifTable* of MIB-II MIB to ‘Up(1)’ or ‘Down(2)’ for the corresponding *ifIndex*. When it is not possible to change the status of a particular interface the switch returns an SNMP error message.

The switch’s implementation of the *ifOperStatus* object in the *ifTable* of MIB-II MIB supports two additional values—“Unknown(4)” and “Dormant(5)” (e.g. an inactive dial-on-demand interface).

Important An unauthorised person with knowledge of the appropriate SNMP community name could bring an interface up or down. Community names act as passwords for the SNMP protocol. Care should be taken when creating an SNMP community with write access to select a secure community name and to ensure that this name is known only to authorised personnel.

Command Reference

This section describes the commands available on the switch to configure and manage the Ethernet, and asynchronous interfaces on the switch.

Some commands require IP and SNMP to be enabled and configured. See [Chapter 13, Internet Protocol \(IP\)](#) for a detailed description of the commands required to enable and configure IP. See [Chapter 39, Simple Network Management Protocol \(SNMP\)](#) for a detailed description of the commands required to enable and configure SNMP.

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 error messages and their meanings.

connect asyn

Syntax `Connect ASYn=0`

Description This command creates a new terminal session that connects a Telnet session directly to a physical asynchronous port. This lets you send commands directly to a device connected to the port. For example, this command can be used to access a modem connected to the port, to send modem commands directly to the modem to change its configuration.

Examples To connect to asynchronous port 0, use the command:

```
connect asy=0
```

Related Commands [connect](#)
[disconnect](#)

disable asyn

Syntax `DISable ASYn=0`

Description This command disables a specific port that is currently enabled so that no data can be accepted or transmitted through it. By default, an asynchronous port is enabled.

If you are logged into an asynchronous port to manage the switch, you cannot disable that port by typing this command. You have to run the command from a script or from a different session, such as a telnet session.

Examples To disable asynchronous port 3, use the command:

```
dis asy=3
```

Related Commands [enable asyn](#)
[show interface](#)
[reset asyn](#)
[set asyn](#)
[show asyn](#)

disable interface linktrap

Syntax `DISable INTERface={ifIndex|interface|DYNAMIC} LInktrap`

where:

- *ifIndex* is a decimal value specifying the entry in the interface MIB
- *interface* is a valid interface name

Description This command disables link up/down trap generation for the specified interface. Link up/down traps are disabled by default.

The **interface** parameter specifies the interface for which link traps are to be disabled.

The **dynamic** parameter handles the special case of dynamic interfaces that do not yet exist. If link traps are enabled for dynamic interfaces, a trap message is generated whenever a dynamic interface is created or destroyed. This is disabled by default. If **dynamic** is specified, link trap generation is disabled for the creation and destruction of dynamic interfaces. Valid interfaces are:

- eth (such as eth0, eth0-1)
- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)

To see a list of current interfaces, use the **show interface** command.

IP and SNMP must be enabled and correctly configured to generate traps. See [Chapter 13, Internet Protocol \(IP\)](#) for a detailed description of the commands required to enable and configure IP. See [Chapter 39, Simple Network Management Protocol \(SNMP\)](#) for a detailed description of the commands required to enable and configure SNMP.

Examples To disable link trap generation for interface vlan1, use the command:

```
dis int=vlan1 li
```

Related Commands [enable interface linktrap](#)
[set interface traplimit](#)
[show interface](#)

enable asyn

Syntax ENABle ASYn=0

Description This command enables a specific asynchronous port. The port must currently be disabled. Data is accepted and/or transmitted via the specified port. By default, an asynchronous port is enabled.

Examples To enable asynchronous port 0, use the command:

```
ena asy=0
```

Related Commands [disable asyn](#)
[reset asyn](#)
[set asyn](#)
[show asyn](#)
[show interface](#)

enable interface linktrap

Syntax ENABle INTerface={ *ifIndex* | *interface* | DYNAMIC } LInktrap

where:

- *ifIndex* is a decimal value specifying the entry in the interface MIB
- *interface* is a valid interface

Description This command enables link up/down traps to be generated for an interface. Link up/down traps are disabled by default.

The **interface** parameter specifies the interface for which link traps are to be enabled.

The **dynamic** parameter handles the special case of dynamic interfaces that do not yet exist. If link traps are enabled for dynamic interfaces, a trap message is generated whenever a dynamic interface is created or destroyed. This is disabled by default. The **dynamic** parameter enables link trap generation for the creation and destruction of dynamic interfaces. Valid interfaces are:

- eth (such as eth0, eth0-1)
- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)

To see a list of current interfaces, use the **show interface** command.

IP and SNMP must be enabled and correctly configured to generate traps. See [Chapter 13, Internet Protocol \(IP\)](#), and [Chapter 39, Simple Network Management Protocol \(SNMP\)](#) for these commands.

Examples To enable link trap generation for the interface with an ifIndex of 1, use the command:

```
ena int=1 li
```

Related Commands [disable interface linktrap](#)
[set interface traplimit](#)
[show interface](#)

purge asyn

Syntax PURge ASYn={*asyn-number*|ALL}

where *asyn-number* is the number of the port. Ports are numbered sequentially starting with 0.

Description This command resets a specific asynchronous port to the factory default configuration. If **all** is specified, all ports are reset and all port configurations are lost.

Examples To purge the configuration of all asynchronous ports, use the command:

```
pur asy=all
```

Related Commands [disable asyn](#)
[enable asyn](#)
[reset asyn](#)
[reset asyn counter](#)
[reset asyn history](#)
[set asyn](#)
[show asyn](#)

reset asyn

Syntax RESET ASYn=*asyn-number*

where *asyn-number* is the number of the port. Ports are numbered sequentially starting with 0.

Description This command resets a specific asynchronous port. If a port number is not specified, then the command applies to the port from which it is issued. If a port number is specified, the command applies to the specified port. The port configuration is restored from nonvolatile storage. Any existing connections are terminated.

Examples To reset asynchronous port 3, use the command:

```
reset asy=3
```

Related Commands [disable asyn](#)
[enable asyn](#)
[reset asyn counter](#)
[reset asyn history](#)
[set asyn](#)
[show asyn](#)

reset asyn counter

Syntax RESET ASYn=*asyn-number* COUNTERS[={Diagnostic|Interface|Rs232}]

where *asyn-number* is the number of the port. Ports are numbered sequentially starting with 0.

Description This command simulates an asynchronous counter reset for the specified asynchronous port. Subsequent [show interface](#) commands, then display only the counter increments since the last **reset asyn** command. SNMP requests however, will still return the counter's true value.

If a port is not specified, then the switch uses the port number from which the command was entered.

If a category is specified, then the command will apply to the counter in that particular category. If a category is not entered, then the command will apply to the counters for all categories. For a description of the categories, see the description of the **show syn** command.

The control signal transition counters displayed by the **show syn** are reset along with the other counters in the RS-232 category.

Examples To reset the interface counter for asynchronous port 3, use the command:

```
reset asy=3 cou=i
```

Related Commands [reset asyn](#)
[reset asyn history](#)
[show asyn](#)

reset asyn history

Syntax RESET ASyn=*asyn-number* History

where *asyn-number* is the number of the port. Ports are numbered sequentially starting with 0.

Description This command clears all commands from the command history for the specified asynchronous port. If a port number is not specified then the command applies to the port or TTY device from which the command is issued. If a port number is specified, the command applies to the specified port.

Port history is automatically reset during the login and logoff processes.

Examples To reset the command history for the asynchronous port to which the terminal is connected, use the command:

```
reset asy h
```

To reset the command history for asynchronous port 3, use the command:

```
reset asy=3 h
```

Related Commands [reset asyn](#)
[reset asyn counter](#)
[show asyn](#)

reset eth

Syntax RESET ETH=*n*

where *n* is the number of the Ethernet interface

Description This command resets a specific Ethernet interface on x900-24X switches

Data being transmitted or received by the Ethernet interface is lost. This may affect the operation of some of the protocol modules in the switch.

Examples To reset Ethernet interface 0, use the command:

```
reset eth=0
```

Related Commands [reset eth counters](#)

reset eth counters

Syntax RESET ETH[=*n*] COunters[={COLlision|DIAGnostic|DOT3stat|INterface}]

where *n* is the number of the Ethernet interface

Description This command simulates an Ethernet counter reset for a specific Eth interface on x900-24X switches. If a number is not specified, the command applies to all Eth interfaces. If a category is specified, the command applies to that particular category. If one is not specified, then the command applies to all categories.

Subsequent **show eth counters** commands then display only the counter increments since the last **reset eth** command. However, SNMP requests still return the counter's true value.

Examples To reset the interface counters for Ethernet interface 0, use the command:

```
reset eth=0 cou=int
```

Related Commands [show eth counters](#)

reset interface counters

Syntax RESET INterface[={*ifIndex*|*interface*}] COunters

where:

- *ifIndex* is a decimal value specifying the entry in the interface MIB
- *interface* is a valid interface

Description This command simulates an interface counter reset. This enables subsequent **show interface** commands to display only the counter increments since the last **reset interface** command. SNMP requests however, still return the counter's true value.

- eth (such as eth0, eth0-1)
- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)

To see a list of current interfaces, use the **show interface** command.

Examples To reset the vlan1 interface MIB counters, use either of the following commands:

```
reset int=vlan1 cou  
reset int=1 cou
```

Related Commands [show interface](#)

set asyn

Syntax for x900-24X

```
SET ASYn[=asyn-number] [Attention={Break|alphabetical control char^[[None}}] [CDcontrol={Connect|Ignore|Online}}] [Databits={5|6|7|8}}] [DTrcontrol={Connect|Off|ON}}] [Echo={ON|OFF|YES|NO|True|False}}] [ENable={BREAK|NONE}}] [Flow={Character|HARdware|None}}] [History=0..99] [IDLEtimeout={10..4294967294|OFF|0}}] [INFlow={Character|HAreware|None}}] [IPaddress={ipadd|None}}] [LOGin={ON|OFF|YES|NO|True|False}}] [MAXoqlen=0..4294967295] [MTu=40..1500] [Name=name] [OUTFlow={Character|HARdware|None}}] [PAGE={0..99|OFF}}] [PARity={Even|Mark|None|Odd|SPace}}] [Prompt={prompt|Default|Off}}] [SECure={ON|OFF|YES|NO|True|False}}] [Shellserver={ON|OFF}}] [SPeed={AUTO|9600|14400|14.4K|19200|19.2K|28800|28.8K|38400|38.4K|57600|57.6K|115200|115.2K}}] [STopbits={1|2}}] [TIMEout=1..65535] [TYpe={Dumb|VT100}}]
```

Syntax for x900-48FE and AT-9900

```
SET ASYn[=asyn-number] [Attention={Break|alphabetical control char^[[None}}] [CDcontrol={Connect|Ignore|Online}}] [Databits={5|6|7|8}}] [DTrcontrol={Connect|Off|ON}}] [Echo={ON|OFF|YES|NO|True|False}}] [ENable={BREAK|NONE}}] [Flow={Character|HARdware|None}}] [History=0..99] [IDLEtimeout={10..4294967294|OFF|0}}] [INFlow={Character|HAreware|None}}] [IPaddress={ipadd|None}}] [IPXnetwork=network] [LOGin={ON|OFF|YES|NO|True|False}}] [MAXoqlen=0..4294967295] [MTu=40..1500] [Name=name] [OUTFlow={Character|HARdware|None}}] [PAGE={0..99|OFF}}] [PARity={Even|Mark|None|Odd|SPace}}] [Prompt={prompt|Default|Off}}] [SECure={ON|OFF|YES|NO|True|False}}] [Shellserver={ON|OFF}}] [SPeed={AUTO|75|110|134.5|150|300|600|1200|1800|2000|2400|4800|9600|14400|14.4K|19200|19.2K|28800|28.8K|38400|38.4K|57600|57.6K|115200|115.2K}}] [STopbits={1|2}}] [TENTimervalue=20..100] [TIMEout=1..65535] [TYpe={Dumb|VT100}}]
```

where:

- *asyn-number* is the number of the port. Ports are numbered sequentially starting with 0.
- *alphabetical control char* is the '^' character followed by any alphabetical character in upper or lower case such as ^A, ^b, ^z.
- *ipadd* is an IP address in dotted decimal notation.
- *network* is a valid Novell network number, expressed as a hexadecimal number. Leading zeros may be omitted.
- *name* is a character string 1 to 15 characters long. If the string contains spaces, it must be in double quotes. The string is not case-sensitive.
- *prompt* is a character string 1 to 15 characters long. If the string contains spaces, it must be in double quotes. The string is not case-sensitive.

Description

This command sets characteristics of asynchronous ports. If a port is not specified, then the command applies to the port on which it is issued. If a port number is specified, the command applies to the specified asynchronous port. Multiple options may be specified in the same command.

If the **set asyn** command is issued from a port with User privileges, the port number and the options **ipaddress**, **mtu**, and **secure** cannot be specified.

For a Telnet connection only, the options **history**, **page**, **prompt**, **type**, and **idletimeout** may be used to alter the behaviour of the dedicated TTY device.

The **set asyn** command may be rejected if there is no hardware present in the switch for the specified port number, the port is currently assigned or a port-pair in a permanent assignment, or the port is a printer port and the printer is active.

The change takes place immediately and the new value is stored in nonvolatile memory.

The **attention** parameter specifies the character used to return from an active session (e.g. a Telnet connection) to the switch prompt. If “^” with an alphabetical character is specified then the attention character is the [Ctrl] key and the specified alphabetical character key held down simultaneously. Similarly, “^[“ means the attention character is set to the [Ctrl] key with the “[“ key. The default is **break** (the [Break] key) for asynchronous ports, and “^P” (the [Ctrl/P] key) for Telnet connections to the switch.

If autobauding is enabled, the attention character must be set to [Break] because this is the only character that can be detected before the baud rate is established

The **cdcontrol** parameter specifies how the switch interprets the state of the DCD input signal. If **cdcontrol** is set to **connect**, when DCD is deasserted, the switch terminates existing connections. This is useful when the port is accessed via a dialup modem. If **cdcontrol** is set to **online**, output is not sent to the port unless the DCD input signal is asserted. When the port is configured as a printer port, and the DTR line of the printer is connected to the DCD input of the switch, the switch determines if a printer is online and powered up. This ensures that print jobs are not sent to a printer that is offline or off. If **cdcontrol** is set to **ignore**, the switch ignores the state of the DCD input regardless of the way the port is used. The default is **ignore**.

The **databits** parameter sets the number of data bits per character transmitted by the port. This should match the terminal setting. The default is **8**.

The **dtrcontrol** parameter controls the way the switch controls the state of the DTR output signal. If **dtrcontrol** is set to **connect**, the DTR output of the switch is asserted for the duration of a valid connection. If **dtrcontrol** is set to **on** or **off**, the DTR line can be driven to the designated state. The default is **on**. This option is intended for ports that are directly connected to host asynchronous ports that require DTR output to be asserted for the duration of a valid connection.

The **echo** parameter sets the echo mode for the port. If **echo** is set to **on**, characters typed following the prompt are echoed to the terminal screen. If **echo** is set to **off**, characters are not echoed to the terminal screen but the switch still receives and processes them. This option has effect when the port is not assigned. When the port is assigned, echoing is controlled by the host. The default is **on**.

The **enable** parameter sets the behaviour of the asynchronous port after you have manually disabled the port by using the [disable asyn command on page 10-17](#). If **enable=break** is specified, you can re-enable the port by sending it a break signal. Further break signals will not affect the port's status. The port remains enabled until it is idle for the **timeout** period, or until you manually re-

enter the **disable asyn** command. See [“Session Timeout” on page 10-12](#) for more information about this functionality. If **enable=none** is specified, the port's status does not change even if it receives a break signal. The default is **break**.

The **flow** parameter sets the flow control mechanism used for the port in both the transmit and receive directions. If **flow** is set to **none**, the switch ignores all incoming flow control characters or lead transitions. The switch does not generate flow control characters and the state of the hardware lines do not change. If **flow** is set to **character**, the switch uses XON/XOFF flow control. If **flow** is set to **hardware**, the switch uses the RTS/CTS lines for flow control. For finer control, the **inflow** and **outflow** parameters can be used to set different flow control mechanisms for the port in the receive and transmit directions, respectively.

The **history** parameter defines the number of commands saved in the command history for future recall with the [show command history command on page 2-14 of Chapter 2, Using the Command Line Interface \(CLI\)](#). The minimum is 0 and the maximum is 99. Setting the history length to zero for a port does not clear all the commands from the history. To clear command history, use the [reset asyn history command on page 10-22](#). The default history length for asynchronous ports and Telnet connections is 30.

The **idletimeout** parameter specifies a period of time, in seconds, for a terminal connection's dedicated TTY device idle timer. If the specified time period lapses since the last time the dedicated TTY device received data from the client, the connection or session is terminated and the terminal screen displays the login prompt. If **0** or **off** are specified, the idle timer remains off, and the session must be explicitly terminated. The default is 0.

If the dedicated TTY device's idle timeout period is modified while there is an established connection, the idle timer for that session is reset so that it uses the new timeout value. Any idle time accumulated by the connection prior to the issuing of the set command is lost.

The **ipaddress** parameter sets the IP address in dotted decimal notation, associated with the port. This parameter may need to be set if the port is used as a network interface using SLIP or PPP. The IP address may be cleared by setting **ipaddress** to **none**. The default is **none**.

The **ipxnetwork** parameter is valid for x900-48FE and AT-9900 switches. It specifies the Novell network number assigned to a user accessing a Novell internetwork via the asynchronous port. The network number may be cleared by setting **ipxnetwork** to **none** instead of a network number. The default is **none**.

The **login** parameter specifies whether a user can log into an asynchronous port and issue commands on the switch. If **on** is specified, users can log into the switch; if **off** is specified, they cannot. No command prompt is displayed, no characters are echoed by the port, and input received by the port is ignored. The default is **on**.



Caution If **login** is set to **off** from a terminal or terminal emulation session over the asynchronous port, it becomes impossible to enter any other commands into that session. In this situation, the switch can be reconfigured from a Telnet session when there is an interface with a valid IP address and appropriate routes. Alternatively, power cycling the switch removes the unsaved configuration.

The **maxoqlen** parameter sets the maximum number of character buffers permitted on the output queue for this port. Once the queue has reached this limit no further buffers are accepted for transmission from the higher layer. The default is **16**. A value of 0 means the length of the output queue is the default value.

The **mtu** parameter sets the Maximum Transmission Unit for the port. This is the maximum number of bytes in a packet transmitted over this port when it is used as a network interface. The minimum MTU is 40 and the maximum is 1500. The default is **1500**.

The **name** parameter assigns a name to the port, as a convenient reference to identify ports. For example, it may be set to the name of the person who normally uses the terminal connected to the port, or the location of the terminal. The default name is "Port #" where "#" is the port number. The name appears in the output of the [show asyn command on page 10-32](#).

The **page** parameter sets the number of lines of command output displayed on the terminal screen before the switch pauses and waits for the user to press a key to continue. This number may range from 0 to 99. The default is **22** for both asynchronous ports and Telnet connections. If **page** is set to **off**, paging is disabled.

The **parity** parameter sets the parity of each character transmitted by the port. This should match the terminal setting. The default is **none**.

The **prompt** parameter sets the prompt for the port to the default string, such as CMD>, or a user-specified string, or it disables the prompt. It is often convenient to disable the prompt when the port is being used as a manager port or for debugging network problems because it reduces the clutter on the terminal screen. This option has effect when the port is not assigned. When the port is assigned, prompting is controlled by the host.

The **secure** parameter determines whether a user must log in to the port before switch commands are accepted. See [Chapter 29, User Authentication](#) for more information on logging in and defining users of the switch. The default is **on** for both asynchronous ports and Telnet connections.

The **shellserver** parameter specifies how to handle characters received on the asynchronous port. Use this parameter to prevent output from a device connected to the port being interpreted as commands. If you specify **on**, characters received on the port are sent to the CLI. If you specify **off**, characters received on the port are ignored. You can still use the [connect command on page 46-7 of Chapter 46, Terminal Server](#) to connect to a device attached to the asynchronous port. The default is **on**.

The **speed** parameter sets the speed (baud rate) of the port. This should match the terminal setting. The attention character must be set to [Break] if autobauding is selected. The port expects to see several [Enter] or [Return] characters to determine the terminal speed setting. If another character is entered initially after the port is reset or cleared, the autobauding feature may not select the correct speed. To restart autobauding in this situation, two consecutive [Break] characters should be entered, followed by two [Enter] or [Return] characters. The default is **auto**.

Autobauding does not work with baud rates exceeding 19200 baud, the maximum for many terminals. A port connected to a modem should not be set to autobauding.

Not all speeds are supported on every switch model. If an unsupported speed is specified, an error message is displayed and the command is ignored.

The **stopbits** parameter sets the number of stop bits per character transmitted by the port. This should match the terminal setting. The default is **1**.

The **tentimervalue** parameter is valid for x900-48FE and AT-9900 switches. It sets the length of the ten timer, in milliseconds. Reducing the length of the ten timer increases the port's responsiveness (see [“Making Asynchronous Ports Respond More Quickly” on page 10-14](#)). Unless you are logged in via the port you want to change, also specify the asynchronous port number. The default **tentimervalue** is 100.

The **timeout** parameter specifies a length of time in seconds for which the asynchronous port can remain idle before it is disabled and the user is logged out. This parameter only takes effect on a port if you have already manually disabled the port using the [disable asyn command on page 10-17](#). To re-enable the port, send it a break signal. See [“Session Timeout” on page 10-12](#) for more information. The **timeout** parameter is only valid if the **enable** parameter is set to **break**. The default timeout is 60 seconds.

The **type** parameter specifies the type of terminal attached to the port. If **type** is set to **vt100**, the switch expects the terminal to support standard VT100 escape sequences and uses them. If **type** is set to **dumb**, the switch does not use VT100 escape sequences. The **dumb** option is usually required for ports connected to printers or very old terminals that do not support VT100 escape sequences. The default is **vt100** for both asynchronous ports and Telnet connections.

Examples The following command configures asynchronous port 17:

```
set asy=17 da=7 par=odd sp=9600 st=1
```

Each parameter can also be set separately:

```
set asy=17 da=7
set asy=17 par=odd
set asy=17 sp=9600
set asy=17 st=1
```

Related Commands [disable asyn](#)
[enable asyn](#)
[reset asyn](#)
[set tty](#)
[show asyn](#)
[show command history](#)
[show tty](#)

set eth speed

Syntax SET ETH=*n* Speed={Autonegotiate|10MHalf|10MFull|100MHalf|100MFull|1000MFull}

where *n* is the number of the Ethernet interface

Description This command sets the link speed for an Ethernet interface on x900-24X switches. The interface number must be specified.

The **speed** parameter specifies the link speed for the interface to use. If **autonegotiate** is specified, the interface negotiates with its Ethernet link partner to choose the fastest speed that both ends of the link can use. The default is **autonegotiate**.



Caution Setting the Ethernet link speed to a value other than the default for the interface may prevent the interface from working properly and may interfere with the operation of other devices on the network. If the interface link speed is set to a value that the link partner does not support, the interface is unable to send or receive data. If the interface link speed is set to a value different from that of the link partner, the **show eth state** command may indicate that the link is up even when it is not.

If the interface link speed is set to **100mhalf** or **100mfull** and the interface is connected to a 10 Mbps network, the interface emits 100 Mbps signals onto the network. This could make the network unusable.

Example To force a 100 Mbps Ethernet interface 0 to operate at 10 Mbps half-duplex regardless of the capabilities of the link partner, use the command:

```
set eth=0 sp=10mh
```

Related Commands [show eth state](#)

set interface mtu

Syntax SET INTERface={*ifIndex*|*interface*} MTU=*value*

where:

- *ifIndex* is a decimal value specifying the entry in the interface MIB
- *interface* is a valid interface
- *value* is an integer Valid values are module-dependent - see below.

Description This command sets the MTU value for the given interface. The **mtu** parameter specifies the value for the maximum transmission unit.

If the interface being set is a PPP interface, and the **mtu** specified in the command is higher than the current MTU on the interface, the PPP connection is reset. This is done so that the higher value may be negotiated through the PPP protocol. If the negotiated rate is lower than the requested MTU, the MTU value is dropped to the negotiated rate. Valid interfaces are:

- eth (such as eth0, eth0-1)
- FR (such as fr0)
- PPP (such as ppp0, ppp1-1)
- ATM (such as atm0.1)
- VLAN (such as vlan1)

To see a list of current interfaces, use the [show interface command on page 10-51](#).

When the MTU is set higher than it is currently, the PPP interface is reset and a short outage on that link may result. The following table lists allowable MTU values.

| Interface | Minimum MTU | Maximum MTU | Default MTU |
|---|-------------|-------------------|-------------|
| PPP (not over Eth) | 256 | 1500 | 1500 |
| PPP over Eth | 256 | 1492 ^a | 1492 |
| VLAN | 256 | 1500 | 1500 |
| The following MTU sizes apply to interfaces that are running Jumbo frames | | | |
| | 256 | 9198 | 1500 |

a. The maximum setting is 8 bytes smaller than the Ethernet interface

Examples To set the MTU value for a PPP3 interface to be 1400, use the command:

```
set int=ppp3 mtu=1400
```

To set the MTU value for an Eth3 interface to be 1400 on an x900-24X, use the command:

```
set int=eth3 mtu=1400
```

Related Commands [show interface](#)

set interface traplimit

Syntax SET INterface={*ifIndex*|*interface*|DYNamic} TRaplimit=1..60

where:

- *ifIndex* is a decimal value specifying the entry in the interface MIB
- *interface* is a valid interface

Description This command sets the maximum number of link up/down traps generated in one minute for the specified interface. The default is 20 trap messages per minute. Valid interfaces are:

- eth (such as eth0, eth0-1)
- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)

To see a list of current interfaces, use the **show interface** command.

IP and SNMP must be enabled and correctly configured to generate traps. See [Chapter 13, Internet Protocol \(IP\)](#) for a detailed description of the commands required to enable and configure IP. See [Chapter 39, Simple Network Management Protocol \(SNMP\)](#) for a detailed description of the commands required to enable and configure SNMP.

Examples To set the trap limit for interface vlan2 to 40, use the command:

```
set int=vlan2 tr=40
```

Related Commands [disable interface linktrap](#)
[enable interface linktrap](#)
[show interface](#)

show asyn

Syntax `SHOW ASyn[=port-number|ALL] [{COUnters[={Diagnostic|
INTErface|Rs232}}]|History|Summary}]`

where *port-number* is the number of the port. Ports are numbered sequentially starting with 0.

Description This command displays configuration information for one or more asynchronous ports. If a port number is specified, then the information for that port is displayed. If a port number is not specified, information for the port from which the command was issued is displayed. If **all** is specified, then information for all the ports on the switch is displayed. If the command is issued from a port with User privilege, the port number may not be specified and the information displayed is for the port from where the command was issued.

If no parameters are specified, then full configuration information for the specified ports is displayed ([Figure 10-1 on page 10-33](#), [Table 10-1 on page 10-34](#)).

The **counter** parameter displays counters from the specified categories ([Figure 10-2 on page 10-37](#), [Table 10-2 on page 10-37](#)). If a category is not specified then counters from all categories are displayed. If **diagnostic** is specified then counters from the asynchronous interface table of the switch enterprise MIB are displayed. If **interface** is specified, then interface counters from the interfaces MIB are displayed. Interface MIB counters exist for ports that are in use as network interfaces. If **rs-232** is specified then counters from the asynchronous port table of the RS-232 like hardware devices MIB are displayed. The **counter** parameter may also not be specified from a port with User privilege.

The **history** parameter displays the command history for the specified ports ([Figure 10-3 on page 10-38](#)). The command history can also be displayed with the [show command history command on page 2-14 of Chapter 2, Using the Command Line Interface \(CLI\)](#). After displaying the command history the switch prompts for a command number from the list. The user can enter a number and press the Enter or Return key to select a command, or just press Enter or Return to return to the prompt. If a valid command number is entered, then the command is displayed at the prompt ready for editing and execution.

The **summary** parameter displays a one-line summary for the specified port or ports ([Figure 10-4 on page 10-38](#), [Table 10-3 on page 10-38](#)).

Figure 10-1: Example output from the **show asyn** command for x900-48FE and AT-9900 switches

```

ASYN 0 : 0000005625 seconds  Last change at: 0000005606 seconds

ASYN information
Name ..... Asyn 0
Status ..... enabled
Mode ..... PPP
Mode ..... Ten
PPP Index ..... 1
TX ACCM ..... 00000000
Data rate ..... 38400
Parity ..... none
Data bits ..... 8
Stop bits ..... 1
Test mode ..... no
In flow state (mode) ..... on  (Hardware)
Out flow state (mode) ..... off (Hardware)
Autobaud mode ..... disabled
Max tx queue length ..... 100
TX queue length ..... 0
Transmit frame ..... none
RX queue length ..... 0
IP address ..... none
Max transmission unit ..... 1500
Ten timer value ..... 100
IPX Network ..... none
Enable Mode.....break
Enabled Status Time Left....59

Control signals
  DTR (out) ..... on  on      1
  RTS (out) ..... on  -       1
  CD  (in)  ..... off connect 0
  CTS (in)  ..... off -       0
  RNG (in)  ..... off -       0

TTY information
Instance ..... 18
Login Name .....
Description ..... Asyn 2
Secure ..... yes
Connections to .....
Current connection ..... none
In flow state ..... on
Out flow state ..... on
Type ..... VT100
Prompt ..... login
Echo ..... yes
Attention ..... break
Manager ..... no
Edit mode ..... insert
History length ..... 20
Page size ..... 22
Idle Timeout (seconds).... 300

```

Table 10-1: Parameters in output of the **show asyn** command

| Parameter | Meaning |
|-----------------------|---|
| Name | The name of the asynchronous port. |
| Status | Whether the port is enabled or disabled. |
| Mode | The mode of operation for the port. This is "Ten" for terminal server ports (characters bundled every tenth of a second). For network interfaces, it is either SLIP, SLIP6, CSLIP, CSLIP6, or SLIPAd. |
| PPP Index | The index for the current PPP session. This field is displayed when the port is being used by ACC for a PPP session. |
| TX ACCM | The current ACCM used by PPP for transmitted control characters. This field is displayed when the port is being used by ACC for a PPP session. |
| Data rate | The baud rate for the port. The default is autobaud. |
| Parity | The parity setting for the port. |
| Data bits | The number of data bits in each transmitted character and the number expected in each received character. |
| Stop bits | The number of stop bits transmitted after each character and the number expected after each received character. |
| Test mode | Whether the interface is in a test mode. |
| In flow state (mode) | The flow control state and mode for the incoming data path. The flow control state may be "on" or "off", indicating whether the port is able to receive characters. The mode may be "none" (no flow control), "hardware" (RTS/CTS flow control), or "XON/XOFF" (XON/XOFF flow control). |
| Out flow state (mode) | The flow control state and mode for the outgoing data path. See "In flow state" for a description. The mode is the same for both directions. |
| Autobaud mode | Whether autobauding is enabled or disabled. When enabled, whether the autobauding process is searching (the port is trying to determine the baud rate of the terminal) or found (the baud rate has been set). |
| Max tx queue length | The maximum number of character buffers permitted on the transmit queue for the port. This parameter affects a port used as a network interface. |
| Tx queue length | The length of the queue of character buffers that are waiting to be transmitted to the port. |
| Transmit frame | The address of the current frame being transmitted by the port, or "none" if no frame is currently being transmitted. |
| Rx queue length | The length of the queue of character buffers that are waiting to be passed up from the port to higher layers. |
| IP address | The IP address set for the port, or "none" if no IP address has been set. |
| Max transmission unit | The maximum number of bytes transmitted in one packet over the interface. |

Table 10-1: Parameters in output of the **show asyn** command (Continued)

| Parameter | Meaning |
|--------------------------|--|
| Ten timer value | The length of the <i>ten timer</i> , in milliseconds. When an asynchronous port is in <i>ten mode</i> , it bundles together the characters that it receives within a certain time period, instead of passing them one at a time to a higher protocol layer for processing. The ten timer sets the time period over which characters are bundled. |
| IPX Network | The IPX network number set for the port, or "none" if no IPX network has been set. |
| Enable Mode | The behaviour of the switch when it receives a break signal. If "break" is displayed, a disabled asynchronous port is enabled when a break signal is received. If "none" is specified, the port's status does not change even if it receives a break signal. |
| Enable Status Time Left | The remaining length of time in seconds for which the asynchronous port can remain inactive before its status is set to disabled. If a timeout time was not specified in the set asyn command, this value is 0. |
| Control signals | The control signals present on the interface, their direction (output or input to the switch), their state and the number of transitions they have made since the switch was powered up or the counters reset. For the DTR and CD signal lines their mode of operation is also displayed. |
| Instance | The instance number for the TTY device dedicated to this port. |
| Login name | The login name of the user logged in to this port, if any. |
| Description | The name assigned to the port. |
| Secure | Whether the port is secure. |
| Connections to | A list of TTY devices (if any) to which this port TTY is linked for the purpose of providing multiple sessions. |
| Current connection | The instance number of the TTY that this port TTY is currently connected to, or "none" if there is no active connection. |
| In flow state | The input flow control state for the TTY dedicated to this port. |
| Out flow state | The output flow control state for the TTY dedicated to this port. |
| Attached module | The module that owns the port. By default this is terminal server. |
| Attached module instance | The instance of the module that owns the port. |
| Type | Whether the terminal type setting for the port is dumb or VT100. |
| Service | The name of the service to which this port belongs, if any. |

Table 10-1: Parameters in output of the **show asyn** command (Continued)

| Parameter | Meaning |
|------------------|---|
| Prompt | Type of prompt given on this port: default off login password confirm encapsulation a user-defined string |
| Echo | Whether the port echoes input characters. |
| Attention | The attention character for this port; either none, break, or char. For an asynchronous port the default attention character is "break". |
| Manager | Whether the port has Manager privilege. |
| Edit mode | The edit mode for the port; either "?", "insert", or "overstrike". The default is "insert". |
| History length | The maximum number of commands that are held in the command history for this port. The default is 30. |
| Page mode/length | The number of lines of command output the switch displays before pausing and waiting for the user to press a key, or "off" when page mode is disabled for this port. The default is 22. |
| Idle Timeout | Maximum period of time in seconds without data being received from a given client before the corresponding session is terminated. |

Figure 10-2: Example output from the **show asyn counter** command

| | | | |
|--|--------|-----------------|--------|
| Asyn 1: 0000014132 seconds Last change at: 0000000000 seconds | | | |
| RS-232 MIB Counters | | | |
| Receive: | | | |
| ParityErrs | 0 | | |
| FramingErrs | 0 | | |
| OverrunErrs | 0 | | |
| Diagnostic Counters | | | |
| Receive: | | Transmit: | |
| inCharacters | 690025 | outCharacters | 689828 |
| inBuffers | 13513 | outBuffers | 13526 |
| fcsErrors | 0 | droppedBuffers | 0 |
| pppErrors | 0 | | |
| slipErrors | 0 | | |
| General: | | | |
| disconnects | 0 | | |
| Interface MIB Counters | | | |
| Receive: | | Transmit: | |
| ifInOctets | 690025 | ifOutOctets | 689828 |
| ifInUcastPkts | 13513 | ifOutUcastPkts | 13526 |
| ifInNUcastPkts | 0 | ifOutNUcastPkts | 0 |
| ifInDiscards | 0 | ifOutDiscards | 0 |
| ifInErrors | 0 | ifOutErrors | 0 |
| ifInUnknownProtos | 0 | ifOutQLen | 0 |

Table 10-2: Parameters in output of the **show asyn counter** command

| Parameter | Meaning |
|----------------|--|
| ParityErrs | Number of characters received with a parity error. |
| FramingErrs | Number of characters received with a framing error. |
| OverrunErrs | Number of characters lost due to an overrun error. |
| inCharacters | Total number of characters received. |
| inBuffers | Number of character buffers transferred to a higher layer. |
| fcsErrors | Number frames received with a frame check sequence error. |
| pppErrors | Number of PPP frames received with errors. |
| slipErrors | Number of SLIP frames received with errors. |
| outCharacters | Total number of characters transmitted. |
| outBuffers | Number of character buffers transmitted for a higher layer. |
| droppedBuffers | Number of character buffers discarded because the output queue had reached its maximum allowed length. |
| disconnects | The number times a SLIP or PPP session has been terminated by the modem disconnecting (dropping CD). |
| ifInOctets | Number of octets received on this interface. |
| ifInUcastPkts | Number of unicast packets delivered to a higher-layer protocol. |
| ifInNUcastPkts | Number of non-unicast packets delivered to a higher-layer protocol. |
| ifInDiscards | Number of inbound packets discarded though no errors had been detected to preventing them from being deliverable to higher-layer protocol. |

Table 10-2: Parameters in output of the **show asyn counter** command (Continued)

| Parameter | Meaning |
|-------------------|--|
| ifInErrors | Number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. |
| ifInUnknownProtos | Number of packets discarded because they were for an unconfigured protocol. |
| ifOutOctets | Number of octets transmitted, including framing. |
| ifOutUcastPkts | Number of unicast packets transmitted or discarded. |
| ifOutNUcastPkts | Number of non-unicast packets transmitted or discarded. |
| ifOutDiscards | Number of packets discarded though no errors had been detected preventing their being transmitted. |
| ifOutErrors | Number of packets not transmitted because of errors. |
| ifOutQLen | Length of the output packet queue. |

Figure 10-3: Example output from the **show asyn history** command

```

1  sh asyn cou
2  sh asyn sum
3  sh asyn hist
4  sh asyn cou
5  login manager
6  sh tty
7  sh asyn=1 cou
8  sh syn cou=int
9  sh asyn=1 cou

Enter command number>

```

Figure 10-4: Example output from the **show asyn summary** command.

```

Port Name          Module Mode    Data Format Attn Secur Mgr
-----
001  Asyn 1        TTY    TTY    19200,N,8,1  brk  yes   no
-----

```

Table 10-3: Parameters in output of the **show asyn summary** command

| Parameter | Meaning |
|-------------|---|
| Port | Number of the asynchronous port. |
| Name | Name assigned to the port. |
| Module | Module that owns the port. |
| Mode | Mode of operation for the port. |
| Data Format | Baud rate, parity, number of data bits and number of stop bits configured for the port. |
| Attn | Attention character for the port; either "-", "brk", or "chr". |
| Secur | Whether the port is secure. |
| Mgr | Whether the port has Manager privilege. |

Examples To show the configuration for asynchronous port 1, use the command:

```
sh asy=1
```

To show all the counters for asynchronous port 1 enter:

```
sh asy=1 cou
```

To see the command history for the asynchronous port to which the terminal is connected enter:

```
sh asy h
```

To obtain an abbreviated display for asynchronous port 1 enter:

```
sh asy=1 s
```

Related Commands

- [disable asyn](#)
- [enable asyn](#)
- [reset asyn](#)
- [reset asyn counter](#)
- [reset asyn history](#)
- [set manager asyn](#)
- [set asyn](#)
- [set tty](#)
- [show tty](#)

show eth configuration

Syntax SHow ETH=*n* CONfiguration

where *n* is the number of the Ethernet interface

Description This command lists modules configured on the x900-24X switch to use an Ethernet interface, and the encapsulations used on an interface ([Figure 10-5](#), [Table 10-4](#)).

Figure 10-5: Example output from the **show eth configuration** command

| | | | | |
|-----------------------------------|----------|----------|---------|--------------|
| Configuration for ETH instance 0: | | | | |
| Module | Protocol | Format | Discrim | MAC address |
| ----- | | | | |
| IPG | IP | Ethernet | 0800 | 0000cd000027 |
| IPG | ARP | Ethernet | 0806 | 0000cd000027 |
| ----- | | | | |

Table 10-4: Parameters in output of the **show eth configuration** command

| Field | Meaning |
|-------------|---|
| Module | Name of the software module that has configured to the Ethernet drivers to receive packets with this encapsulation. |
| 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 packets that should receive the specified format. |
| MAC Address | Media Access Control source address for which the module wishes to receive packets. This is commonly known as the Ethernet address. |

Related Commands [show eth counters](#)
[show eth receive](#)

show eth counters

Syntax `SHoW ETH[=n] COUnTERS[={COLlision|DIAGnostic|DOT3stat|INTErface}]`

where *n* is the number of the Ethernet interface

Description This command displays MIB counters for an Ethernet interface on x900-24X switches. If the interface is not specified, the counters for all Ethernet interfaces are displayed. If a category is not specified, all counters are displayed.

If **collision** is specified, collision statistics counters from the dot3 MIB are displayed (Figure 10-6 on page 10-41). If the Ethernet interface hardware does not report collision statistics the collision frequency values are all displayed as zero. Collision frequencies are displayed in pairs of columns, representing a histogram. The right hand column (the value axis of the histogram) is a count of individual MAC frames for which the transmission on the specified interface was accompanied by the number of per-frame media collisions specified in the left hand column (the category axis of the histogram).

If **diagnostic** is specified, diagnostic counters specific to the switch's Ethernet hardware are displayed. The output is divided into two parts, *device independent* counters that are the same for all switch models, and *device dependent* counters that differ depending on whether the interface uses 68360 hardware (Figure 10-6 on page 10-41), SONIC hardware (Figure 10-7 on page 10-42), M860F hardware (Figure 10-8 on page 10-42), or 79C972 hardware (Figure 10-9 on page 10-43).

Table 10-5 on page 10-43 has descriptions of parameters.

If **dot3stat** is specified, statistics counters from the dot3 MIB are displayed (Figure 10-10 on page 10-45, Table 10-6 on page 10-45).

If **interface** is specified, interface counters from the MIB-II MIB are displayed (Figure 10-11 on page 10-46, Table 10-7 on page 10-46).

Figure 10-6: Example output from the **show eth counters=collision** command

| | | | | | | | |
|--|----|--------------|-----------------|-----|-----------|-----|---|
| ETH instance 0: | | 1245 seconds | Last change at: | | 0 seconds | | |
| dot3 MIB Collision Statistics Counters | | | | | | | |
| Collision frequencies: | | | | | | | |
| 1: | 11 | 5: | 0 | 9: | 0 | 13: | 0 |
| 2: | 9 | 6: | 0 | 10: | 0 | 14: | 0 |
| 3: | 3 | 7: | 0 | 11: | 0 | 15: | 0 |
| 4: | 0 | 8: | 0 | 12: | 0 | 16: | 0 |

Figure 10-7: Example output from the **show eth counters=diagnostic** command for 68360-based hardware

| | | | |
|---|-------------|----------------------|-----------|
| ETH instance 0: | 438 seconds | Last change at: | 0 seconds |
| Device Independent Diagnostic Counters (68360 hardware) | | | |
| EthProtoCacheHit | 463 | EthProtoCacheMiss | 58 |
| DSAPProtoCacheHit | 0 | DSAPProtoCacheMiss | 0 |
| SNAPProtoCacheHit | 0 | SNAPProtoCacheMiss | 0 |
| RxFIFOOverrun | 0 | TxFIFOUnderrun | 0 |
| RxTooFewBuffers | 0 | TxTooManyFragments | 0 |
| BusError | 0 | TxDescriptorAreaFull | 0 |
| Reset | 0 | TxFrameTooLong | 0 |
| LoadCAMFailure | 0 | TxLostInterrupt | 0 |
| Device Dependent Diagnostic Counters (68360 hardware) | | | |
| CommandTimeout | 0 | TxNoPacket | 0 |
| Command | 0 | | |

Figure 10-8: Example output from the **show eth counters=diagnostic** command for M860F-based hardware

| | | | |
|---|------------|----------------------|-----------|
| ETH instance 0: | 21 seconds | Last change at: | 2 seconds |
| Device Independent Diagnostic Counters | | | |
| EthProtoCacheHit | 203 | EthProtoCacheMiss | 26 |
| DSAPProtoCacheHit | 0 | DSAPProtoCacheMiss | 0 |
| SNAPProtoCacheHit | 0 | SNAPProtoCacheMiss | 0 |
| RxFIFOOverrun | 0 | TxFIFOUnderrun | 0 |
| RxTooFewBuffers | 0 | TxTooManyFragments | 0 |
| BusError | 0 | TxDescriptorAreaFull | 0 |
| Reset | 0 | TxFrameTooLong | 0 |
| LoadCAMFailure | 0 | TxLostInterrupt | 0 |
| Device Dependent Diagnostic Counters (M860F hardware) | | | |
| LinkChanges | 1 | MIITimeouts | 0 |
| AutoNegCompletes | 1 | ParallelDetFaults | 0 |

Figure 10-9: Example output from the **show eth counters=diagnostic** command for 79C972-based hardware

| | | | |
|--|------------|-------------------------|-----------|
| ETH instance 1: | 35 seconds | Last change at: | 1 seconds |
| Device Independent Diagnostic Counters | | | |
| EthProtoCacheHit | 212 | EthProtoCacheMiss | 40 |
| DSAPProtoCacheHit | 0 | DSAPProtoCacheMiss | 0 |
| SNAPProtoCacheHit | 0 | SNAPProtoCacheMiss | 0 |
| RxFIFOOverrun | 0 | TxFIFOUnderrun | 0 |
| RxTooFewBuffers | 0 | TxTooManyFragments | 0 |
| BusError | 0 | TxDescriptorAreaFull | 0 |
| Reset | 0 | TxFrameTooLong | 0 |
| LoadCAMFailure | 0 | TxLostInterrupt | 0 |
| Device Dependent Diagnostic Counters (79C972 hardware) | | | |
| LinkChanges | 1 | ReceiveCollisions | 0 |
| AutoNegCompletes | 0 | ParallelDetFaults | 0 |
| MIIPhyDetectTransitions | 0 | MIIManagementReadErrors | 0 |

Table 10-5: Parameters in output of the **show eth counters=diagnostic** command

| Counter | Meaning |
|--------------------|--|
| EthProtoCacheHit | The number of times for an Ethernet protocol packet the Ethernet type field matched that of a protocol discriminator structure in the cache. |
| DSAPProtoCacheHit | The number of times for a DSAP protocol packet the DSAP field matched that of a protocol discriminator structure in the cache. |
| SNAPProtoCacheHit | The number of times for a SNAP protocol packet the SNAP field matched that of a protocol discriminator structure in the cache. |
| RxFIFOOverrun | The number of times reception of a packet failed due to a FIFO overrun. |
| RxTooFewBuffers | The number of times that after the reception of a packet or during recovery from a receive buffers exhausted interrupt there were insufficient free buffers to replenish the queue of buffers available for reception. |
| BusError | The number of times a direct memory access transfer was aborted due to a bus error. |
| Reset | The number of times the ETHRecover routine was called in response to a serious error. |
| LoadCAMFailure | The number of times a load of the CAM failed. |
| EthProtoCacheMiss | The number of times for an Ethernet protocol packet the Ethernet type field matched that of a protocol discriminator structure in the discriminator list but the structure was not in the cache. |
| DSAPProtoCacheMiss | The number of times for a DSAP protocol packet the DSAP field matched that of a protocol discriminator structure in the discriminator list but the structure was not in the cache. |

Table 10-5: Parameters in output of the **show eth counters=diagnostic** command (Continued)

| Counter | Meaning |
|-------------------------|--|
| SNAPProtoCacheMiss | The number of times for a SNAP protocol packet the SNAP field matched that of a protocol discriminator structure in the discriminator list but the structure was not in the cache. |
| TxFIFOUnderrun | The number of times the transmission of a packet failed due to a FIFO underrun. |
| TxTooManyFragments | The number of times a packet could not be transmitted because it contained too many fragments. |
| TxDescriptorAreaFull | The number of times there was insufficient room in the Transmit Descriptor Area because there were so many packets queued for transmission and not yet transmitted. |
| TxFrameTooLong | The number of times a frame was not transmitted because it exceeded the maximum length of an Ethernet frame. |
| TxLostInterrupt | The number of times the lost transmit interrupt timer timed out before a packet had been transmitted. |
| CommandTimeout | [68360 hardware] The number of times a command to the Ethernet hardware did not complete before the timeout timer expired. |
| Command | [68360 hardware] The code of the command that was to be issued when a command timeout was detected. |
| TxNoPacket | [68360 hardware] The number of times the Ethernet hardware reported a transmit error, but there was no packet being transmitted or the errored packet could not be identified. |
| LinkChanges | (M860F and 79C972 hardware) The number of times the Ethernet link status (up, down, or unknown) changed. |
| AutoNegCompletes | (M860F and 79C972) The number of times the Ethernet hardware completed negotiating a set of operating parameters with the link partner (a hub or switch) to which it is connected. |
| ParallelDetFaults | (M860F and 79C972 hardware) The number of times the Ethernet hardware reported a parallel detection fault while negotiating with its link partner. |
| MIITimeouts | [M860F hardware] The number of times a command from the Ethernet hardware's controller to its transceiver did not complete before the timeout timer expired. |
| MIIPhyDetectTransitions | [79C972 hardware] The number of times the Ethernet hardware's controller detected changes in the presence of the transceiver. |
| MIIManagementReadErrors | [79C972 hardware] The number of times the Ethernet hardware detected an error in communications between its controller and transceiver. |
| ReceiveCollisions | [79C972 hardware] The number of times the Ethernet hardware detected a late collision on the network while receiving data. |

Figure 10-10: Example output from the **show eth counters=dot3stat** command

| | | | |
|--|------|-------------------------|----|
| ETH instance 0: 1295 seconds Last change at: 0 seconds | | | |
| dot3 Statistics MIB Counters | | | |
| Receive: | | Transmit: | |
| InternalMacRxErrors | 0 | InternalMacTxErrors | 0 |
| FrameTooLongs | 0 | DeferredTransmissions | 5 |
| AlignmentErrors | 0 | SingleCollisionFrames | 11 |
| FCSErrors | 2 | MultipleCollisionFrames | 12 |
| Missed | 0 | LateCollisions | 0 |
| UnwantedBroad | 5114 | ExcessiveCollisions | 0 |
| UnwantedMulticasts | 5 | CarrierSenseErrors | 0 |
| RxQueueLength | 0 | ExcessiveDeferrals | 0 |

Table 10-6: Parameters in output of the **show eth counters=dot3stat** command

| Counter | Meaning |
|-------------------------|--|
| InternalMacRxErrors | Number of frames for which reception failed due to an internal error. |
| FrameTooLongs | Number of frames received that exceeded the maximum permitted frame size. |
| AlignmentErrors | Number of received frames with alignment and FCS errors. |
| FCSErrors | Number of received frames with FCS but not alignment errors. |
| Missed | Number of packets not received due to: (1) lack of memory resources to buffer the packet, (2) a FIFO overrun, (3) receiver was disabled. |
| UnwantedBroad | Number of broadcast frames received on this interface for a protocol not configured to any module. |
| UnwantedMulticasts | Number of multicast frames received on this interface for a protocol not configured to any module. |
| RxQueueLength | Length of the queue of receive packets between the interrupt routine and the idle level receive packet processing routine. |
| InternalMacTxErrors | Number of frames not transmitted due to internal error. |
| DeferredTransmissions | Number of frames delayed by busy medium. |
| SingleCollisionFrames | Number of successfully transmitted frames that were inhibited by exactly one collision. |
| MultipleCollisionFrames | Number of successfully transmitted frames that were inhibited by more than one collision. |
| LateCollisions | Number of times that a collision is detected later than 512 bit times into the transmission of a packet. |
| ExcessiveCollisions | Number of frames not transmitted due to excessive collisions. |
| CarrierSenseErrors | Number of times that carrier sense was lost or never asserted during the transmission of a frame. |
| ExcessiveDeferrals | Number of times transmission of a packet was aborted due to excessive deferrals. |

Figure 10-11: Example output from the **show eth counters=interface** command

| | | | |
|------------------------------|---------|---------------------------|--------|
| ETH instance 0: 1239 seconds | | Last change at: 0 seconds | |
| Interface MIB Counters | | | |
| Receive: | | Transmit: | |
| ifInOctets | 2357609 | ifOutOctets | 872296 |
| ifInUcastPkts | 2588 | ifOutUcastPkts | 3985 |
| ifInNUcastPkts | 420 | ifOutNUcastPkts | 2 |
| ifExtnsMulticastsRxOKs | 5 | ifExtnsMulticastsTxOKs | 0 |
| ifExtnsBroadcastsRxOKs | 5308 | ifExtnsBroadcastsTxOKs | 2 |
| ifInDiscards | 0 | ifOutDiscards | 0 |
| ifInErrors | 2 | ifOutErrors | 0 |
| ifInUnknownProtos | 4888 | ifOutQLen | 1 |

Table 10-7: Parameters in output of the **show eth counters=interface** command

| Counter | Meaning |
|---------------------------------|---|
| ifLastChange | Value of sysUpTime at the time the interface entered its current operational state. |
| ifInOctets | Number of octets received on this interface. |
| ifInUcastPkts | Number of unicast packets delivered to a higher-layer protocol. |
| ifInNUcastPkts | Number of non-unicast packets delivered to a higher-layer protocol. |
| ifExtnsMulticastsReceivedOKs | Number of frames successfully received for a multicast address other than a broadcast address. |
| ifExtnsBroadcastsReceivedOKs | Number of frames successfully received for a broadcast address other than a multicast address. |
| ifInDiscards | Number of inbound packets discarded without errors that prevented them from being deliverable to higher-layer protocol. |
| ifInErrors | Number of inbound packets with errors that prevented them from being deliverable to a higher-layer protocol. |
| ifInUnknownProtos | Number of packets discarded because they were for an unconfigured protocol. |
| ifOutOctets | Number of octets transmitted, including framing. |
| ifOutUcastPkts | Number of unicast packets transmitted or discarded. |
| ifOutNUcastPkts | Number of non-unicast packets transmitted or discarded. |
| ifExtnsMulticastsTransmittedOKs | Number of frames successfully transmitted to a multicast address other than a broadcast address. |
| ifExtnsBroadcastsTransmittedOKs | Number of frames successfully transmitted to a broadcast address other than a multicast address. |
| ifOutDiscards | Number of packets discarded though no errors had been detected preventing their being transmitted. |
| ifOutErrors | Number of packets not transmitted because of errors. |
| ifOutQLen | Length of the output packet queue. |

Related Commands

- [reset eth counters](#)
- [show eth configuration](#)
- [show eth receive](#)

show eth macaddress

Syntax `SHoW ETH[=n] MACaddress`

where *n* is the number of the Ethernet interface

Description This command displays the default MAC address for a specific Ethernet interface. If the interface is not specified, default MAC addresses for all Ethernet interfaces are displayed ([Figure 10-12](#)).

Figure 10-12: Example output from the **show eth macaddress** command

```
MAC address for ETH instance 0:

Address
-----
00-00-cd-00-0d-0e
-----
```

Related Commands

- [show eth configuration](#)
- [show eth counters](#)
- [show eth receive](#)

show eth receive

Syntax SHow ETH[=*n*] RECeive

where *n* is the number of the Ethernet interface

Description This command displays the multicast addresses that an Ethernet interface has been configured to receive on x900-24X switches. If the interface is not specified, the multicast addresses for all Ethernet interfaces are displayed (Figure 10-13).

Note that the list includes the broadcast address and any unicast addresses specified by the software modules that have configured to the Ethernet interface. Unicast addresses are distinguishable from multicast addresses by their first octet. The first octet of a unicast address is even, whereas for a multicast address it is odd. The broadcast address is a special multicast address that is received by all stations on an Ethernet. The switch is always configured to receive broadcast packets, even if no software modules are using the interface, so the list always includes the broadcast address.

Figure 10-13: Example output from the **show eth receive** command

```
Receive addresses for ETH instance 0:
```

```
Address
```

```
-----  
00-00-cd-00-0d-0e
```

```
ff-ff-ff-ff-ff-ff  
-----
```

Related Commands [show eth configuration](#)
[show eth counters](#)

show eth state

Syntax `SHoW ETH[=n] STaTe`

where *n* is the number of the Ethernet interface

Description On x900-24X switches, this command displays the state of the link between the Ethernet interface and its link partner (the Ethernet hub or switch to which it is connected). If an Ethernet interface is not specified, information about all Ethernet interfaces is displayed.

The information displayed depends on the capabilities of the Ethernet interface hardware and how the interface is configured (Figure 10-15, Table 10-8 on page 10-49).

Figure 10-14: Example output from the **show eth state** command for 100Mbps interfaces in the default configuration (with link speed auto-negotiation enabled)

```
State for ETH instance 0:

Configured speed/duplex..... 100 Mbps, half duplex
Actual speed/duplex ..... 10 Mbps, half duplex
Duplex mode ..... full
Auto-negotiation ..... complete

Link partner capabilities
Auto-negotiation ..... yes
100BASE-TX full duplex ..... yes
100BASE-TX ..... yes
10BASE-T full duplex ..... yes
10BASE-T ..... yes
```

Figure 10-15: Example output from the **show eth state** command for 100Mbps interfaces with link speed set manually

```
State for ETH instance 0:

Link ..... up
Speed ..... 100 Mbps
Duplex mode ..... half
Auto-negotiation ..... disabled
```

Table 10-8: Parameters in output of the **show eth state** command

| Parameter | Meaning |
|-------------------------|---|
| Configured speed/duplex | The port speed mode configured for this port. Either "Autonegotiate" or a combination of a speed (either 10 Mbps, 100 Mbps, or 1000 Mbps) and a duplex mode (half duplex or full duplex) and optionally by autonegotiation. |
| Actual speed/duplex | The port speed and duplex mode that this port is actually running at. A combination of a speed (either 10 Mbps, 100 Mbps, or 1000 Mbps) and a duplex mode (half duplex or full duplex). |
| Link | The current state of the link; one of up or down. |
| Speed | The speed at which the link is operating. |

Table 10-8: Parameters in output of the **show eth state** command (Continued)

| Parameter | Meaning |
|---------------------------|--|
| Duplex mode | Whether the duplex mode in which the link is operating is half duplex or full duplex. |
| Auto-negotiation | Whether the negotiation process is in progress or complete between the Ethernet interface and its link partner to set link parameters. |
| Link partner capabilities | Information about the capabilities of the link partner. |
| Auto-negotiation | Whether the link partner can automatically negotiate link parameters. |
| 100BASE-TX full duplex | Whether the link partner can operate at 100 Mbps full-duplex. |
| 100BASE-TX | Whether the link partner can operate at 100 Mbps half-duplex. |
| 10BASE-TX full duplex | Whether the link partner can operate at 10 Mbps full-duplex. |
| 10BASE-TX | Whether the link partner can operate at 10 Mbps half-duplex. |

Examples To show the state of Ethernet interface 1, use the command:

```
sh eth=1 sta
```

Related Commands [set eth speed](#)

show interface

Syntax `SHoW INTeRface[={ifIndex|interface}] [COUnters]`

where:

- *ifIndex* is a decimal value specifying the entry in the interface MIB
- *interface* is a valid interface name

Description This command displays the contents of the interface MIB. If an interface is not specified, summary information for all interfaces is displayed (Figure 10-16, Table 10-9 on page 10-51). If an interface is specified, detailed information is displayed about it including the counters (Figure 10-17 on page 10-52, Table 10-10 on page 10-52).

The **counters** parameter displays interface counters for all interfaces (Figure 10-18 on page 10-53, Table 10-11 on page 10-53). When the interface is a VLAN, the command displays counters for packets switched by the CPU, not those switched in hardware at wire speed.

Figure 10-16: Example output from the **show interface** command

```

Interfaces                                sysUpTime:                                03:45:33

DynamicLinkTraps.....Disabled
TrapLimit.....20

ifIndex Interface ifAdminStatus ifOperStatus ifLastChange
-----
1      port1      Up      Down      00:00:16
2      port2      Up      Down      00:00:00
3      port3      Up      Down      00:00:00
4      port4      Up      Down      00:00:00
5      port5      Up      Down      00:00:00
6      port6      Up      Down      00:00:00
7      port7      Up      Down      00:00:00
8      port8      Up      Down      00:00:00
9      port9      Up      Down      00:00:00
10     port10     Up      Down      00:00:00
11     port11     Up      Down      00:00:00
12     port12     Up      Down      00:00:00
13     port13     Up      Down      00:00:00
14     port14     Up      Down      00:00:00
15     port15     Up      Down      00:00:00
16     port16     Up      Down      02:21:50
17     vlan1      Up      Down      02:21:50
-----

```

Table 10-9: Parameters in output of the **show interface** command

| Parameter | Meaning |
|------------------------------------|---|
| sysUpTime | Elapsed time since the last switch restart. |
| DynamicLinkTraps | Whether link traps are enabled for dynamic interfaces. |
| TrapLimit | Maximum number of link up/down traps for dynamic interfaces that is generated in one minute. |
| Number of unencrypted PPP/FR links | Total number of PPP interfaces that are configured to send plaintext. Does not include disabled PPP interfaces. |

Table 10-9: Parameters in output of the **show interface** command (Continued)

| Parameter | Meaning |
|---------------|---|
| ifIndex | Index of the interface in the interface table. |
| Interface | Name of the interface. |
| ifAdminStatus | Whether the administratively-set (configured) state of the interface is Up, Down, or Testing. |
| ifOperStatus | Whether the operational state of the interface is Up, Down, Testing, Unknown, or Dormant. |
| ifLastChange | Value of <i>sysUpTime</i> at the time the interface entered its current operational state. |

Figure 10-17: Example output from the **show interface** command for a specific interface

```

Interface..... vlan1
  ifIndex..... 1
  ifMTU..... 1500
  ifSpeed..... 10000000
  ifAdminStatus..... Up
  ifOperStatus..... Up
  ifLinkUpDownTrapEnable... Disabled
  TrapLimit..... 20

Interface Counters

  ifInOctets ..... 21484      ifOutOctets ..... 13775
  ifInUcastPkts ..... 165      ifOutUcastPkts ..... 134
  ifInNUcastPkts ..... 19      ifOutNUcastPkts ..... 0
  ifInDiscards ..... 0         ifOutDiscards ..... 0
  ifInErrors ..... 0           ifOutErrors ..... 0
  ifInUnknownProtos ... 30

```

Table 10-10: Parameters in output of the **show interface** command for a specific interface

| Parameter | Meaning |
|------------------------|---|
| Interface | Name of the interface. |
| ifIndex | Index of the interface in the interface table. |
| ifMTU | Size in octets of the largest packet that can be transmitted on the interface. |
| ifSpeed | Estimate of the interface's current speed in bits per second, or 0 if the interface is down. |
| ifAdminStatus | Whether the administratively-set (configured) state of the interface is Up, Down, or Testing. |
| ifOperStatus | Whether the operational state of the interface is Up, Down, Testing, Unknown, or Dormant. |
| ifLinkUpDownTrapEnable | Whether link traps are enabled for the interface. |
| TrapLimit | Maximum number of link up/down traps for dynamic interfaces that is generated in one minute. |
| Interface Counters | Counters for the interface. |
| ifInOctets | Number of octets (bytes) received by the interface. |
| ifInUcastPkts | Number of unicast packets received by the interface. |

Table 10-10: Parameters in output of the **show interface** command for a specific interface (Continued)

| Parameter | Meaning |
|-----------------|--|
| ifInNUcastPkts | Number of multicast packets received by the interface. |
| ifInDiscards | Number of packets discarded by the interface. Not applicable for a port interface. |
| ifInErrors | Number of packets received with errors by the interface. |
| ifUnknownProtos | Number of packets received by the interface but discarded because their protocol is unsupported. |
| ifOutOctets | Number of bytes transmitted by the interface. |
| ifOutUcastPkts | Number of unicast packets transmitted by the interface. |
| ifOutNUcastPkts | Number of multicasts transmitted by the interface. |
| ifOutDiscards | Number of output packets discarded by the interface. Not applicable for a port interface. |
| ifOutErrors | Number of packets that should have been transmitted but were not because of errors. |

Figure 10-18: Example output from the **show interface counter** command

| | | | |
|--------------------|-------------|------------------|-------------|
| Interface Counters | | | |
| Interface: vlan1 | | | |
| ifInOctets | 22852 | ifOutOctets | 15565 |
| ifInUcastPkts | 184 | ifOutUcastPkts | 148 |
| ifInNUcastPkts | 19 | ifOutNUcastPkts | 0 |
| ifInDiscards | 0 | ifOutDiscards | 0 |
| ifInErrors | 0 | ifOutErrors | 0 |
| ifInErrors | 0 | ifOutErrorOctets | 0 |
| ifInUnknownProtos | ... 30 | | |

Table 10-11: Parameters in output of the **show interface counter** command

| Parameter | Meaning |
|-----------------|--|
| Interface | Name of the interface. |
| ifInOctets | Number of octets (bytes) received by the interface. |
| ifInUcastPkts | Number of unicast packets received by the interface. |
| ifInNUcastPkts | Number of multicast packets received by the interface. |
| ifInDiscards | Number of packets discarded by the interface. Not applicable for a port interface. |
| ifInErrors | Number of packets received with errors by the interface. |
| ifUnknownProtos | Number of packets received by the interface but discarded because their protocol is unsupported. |
| ifOutOctets | Number of bytes transmitted by the interface. |
| ifOutUcastPkts | Number of unicast packets transmitted by the interface. |
| ifOutNUcastPkts | Number of multicasts transmitted by the interface. |
| ifOutDiscards | Number of output packets discarded by the interface. Not applicable for a port interface. |
| ifOutErrors | Number of packets that should have been transmitted but were not because of errors. |

Examples To display the general state of all interfaces, use the command:

```
sh int
```

Related Commands [disable interface linktrap](#)
[enable interface linktrap](#)
[set interface traplimit](#)